

**Universidad Tecnológica Nacional**

Proyecto Final

---

**Sistema de control para playas de  
estacionamiento**

---

*Autores:*

- Lissa, Marcelo Eduardo
- Wagner, Emiliano Nicolás

*Director: Ing. Dachary, Alejandro*

*Proyecto final presentado para cumplimentar los requisitos académicos  
para acceder al título de Ingeniero en Electrónica*

*en la*

**Facultad Regional Paraná**

Noviembre de 2017



## **Declaración de autoría:**

Nosotros declaramos que el Proyecto Final “Sistema de control para playas de estacionamiento” y el trabajo realizado son propios. Declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero en Electrónica, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

- 
- 

Fecha:



## Agradecimientos:

Al ing. Alejandro Dachary por habernos brindado la idea, aconsejado y por estar siempre dispuesto a ayudarnos a poder realizar este proyecto.

En primera medida quiero agradecer a Dios por todas las oportunidades que me brindo a lo largo de mi vida. A mis padres, por toda la confianza y el apoyo que me dieron desde el primer momento. También quiero agradecer a mi hermano por sus consejos y por las veces que me ayudó con el estudio de las materias. A los compañeros y amigos que hice a lo largo de la carrera, por todos los buenos momentos que pasamos y por hacer que el vivir lejos de casa no sea tan difícil y finalmente a Nico, por haber aceptado ser mi compañero de proyecto.

Lissa, Marcelo Eduardo.

En primer lugar quiero agradecer a Dios por todo lo que me ha dado, a mis padres por darme la vida, su amor y esfuerzo para poder llegar a este momento. También a mis hermanos y a mi hermana que me bancaron todo el tiempo. A mis amigos de Bovril, con los que tuve la suerte de poder cursar y vivir en esta ciudad, y a mis amigos de la facultad con los cuales pasamos todos estos hermosos años cursando y compartiendo momentos.

Wagner, Emiliano Nicolás.



Universidad Tecnológica Nacional

## *Abstract*

Facultad Regional Paraná

Ingeniero en Electrónica

# Sistema de control para playas de estacionamiento

Lissa, Marcelo Eduardo

Wagner, Emiliano Nicolás

### **Abstract:**

This Project developed an automation system for parking lots. The system allows the sensing of vehicles, signage of the different places within the parking lot and digital processing of images for the detection of plates, which is an innovation at a national level.

In order to do it the OpenCV library was used to process images and identify the plates that enter or leave the parking lots, the images of the vehicles are obtained through IP cameras. The RS-485 protocol is used for communication of all units and the HC-SR04 ultrasonic sensor is used for the detection of parked vehicles. The MPLABX programming environment of Microchip was used for the programming of the microcontrollers and Microsoft Visual Studio 2017 was used for the development of the PC application.

When the project was finalized was possible to detect Argentine plates with the Mercosur and 1995 formats when they are between 1 and 3 [m] of the entry/exit booth. It was possible to keep a record of the vehicles that entered the parking, the free and occupied places inside the parking,

calculate the time that a car was inside and the amount to be charged, all this done automatically. The operator must only be present to collect the service.

**Keywords:**

Communication, Digital Image Processing, OpenCV, Plate, Vehicle Detection.

**Resumen:**

En el presente proyecto se realizó un sistema de automatización para playas de estacionamiento que utiliza el procesamiento digital de imágenes para la detección de patentes y la señalización del estado de disponibilidad en cada lugar como innovación a nivel nacional.

Para llevarlo a cabo se utilizó la librería OpenCV para procesar las capturas de las patentes que ingresan o egresan de la playa a través de cámaras IP nativas, el protocolo RS-485 para la comunicación de todas las unidades, el sensor ultrasónico HC-SR04 para la detección de los vehículos estacionados, el entorno de programación MPLABX de Microchip para el desarrollo del firmware y Microsoft Visual Studio 2017 para el de la aplicación de PC.

Una vez finalizado se logró detectar las patentes argentinas con el formato de 1995 y del Mercosur a una distancia entre 1 y 3[m], llevar un registro de los vehículos ingresados, de los lugares libres y ocupados dentro de la playa, calcular el tiempo de uso de un determinado auto y el monto a cobrar, todo esto realizado de manera automática quedando como única tarea del operario el cobrar el servicio.

**Palabras Clave:**

Comunicación, Detección de Vehículos, OpenCV, Patentes, Procesamiento Digital de Imágenes.

## *Reconocimientos:*

Al Ing. Alejandro Dachary quien además de brindarnos la idea para el proyecto, nos facilitó equipos e información.

A nuestros amigos Martín Acosta, Milton Sigura, Ariel Parodi, Javier Romero, Facundo Beade, Ezequiel Tello De Meneses, Andrés Pagura, Emiliano Estrubia y Enzo Longo por la ayuda que nos dieron, tanto prestando herramientas, como por los consejos dados.

# Índice:

Capítulo 1. Introducción .....	18
Capítulo 2. Desarrollo .....	20
2.1 Hardware.....	20
2.1.1 Diagrama de Bloques .....	20
2.1.2 Unidad Remota (UR) .....	20
2.1.3 Problemas y soluciones implementadas.....	26
2.1.4 BUS de comunicaciones .....	27
2.1.5 Problemas y soluciones implementadas.....	29
2.1.6 Unidad Central (UC) .....	29
2.1.7 Problemas y soluciones implementadas.....	33
2.1.8 Cartel de entrada .....	34
2.2 Procesamiento Digital de Imágenes.....	37
2.2.1 Diagrama de bloques .....	37
2.2.2 OpenCV .....	38
2.2.3 Detección de patentes .....	38
2.2.4 Pruebas realizadas .....	51
2.2.5 Cámara .....	53
2.3 Aplicación de PC.....	56
2.3.1 Pruebas de la Aplicación de PC.....	61
2.3.2 Problemas y Soluciones de la Aplicación de PC .....	66
2.4 Diseño Completo (Prototipo) .....	67
Capítulo 3. Resultados .....	71
Capítulo 4. Análisis de Costos.....	72
Capítulo 5. Discusión y Conclusión .....	74
Capítulo 6. Literatura Citada.....	76

## Lista de Figuras:

Figura 1: Sensor ultrasónico HC-SR04.....	21
Figura 2: Circuito del relé.....	22
Figura 3: Lámparas tipo gota LED.....	22
Figura 4: Fuente de alimentación de una UR.....	23
Figura 5: Conexión típica Half-Duplex RS-485.....	28
Figura 6: Transceptor MAX487.....	28
Figura 7: Fuente de alimentación de la UC.....	30
Figura 8: Decodificador de HEF4028B.....	34
Figura 9: Circuito esquemático del bloque de LEDs.....	35
Figura 10: Patente argentina desde 1995 a 2016.....	39
Figura 11: Patente del Mercosur desde 2016 hasta la actualidad.....	39
Figura 12: a) Imagen original, b) Imagen luego del filtro sobel, c) Imagen luego del filtro umbral, d) Imagen luego de operación morfológica, e) Máscara de zona de llenado f) Patente segmentada, g) Caracteres detectados.....	41
Figura 13: Ejemplo de uso de SVM.....	44
Figura 14: Imágenes filtradas de patentes para detectar sus caracteres.....	47
Figura 15: Ejemplo de perceptrón de 3 capas.....	48
Figura 16: Funcionamiento general de una neurona.....	49
Figura 17: Tipografía Schriften.....	50
Figura 18: Configuración de imagen de la cámara IP.....	54
Figura 19: Pestaña de Inicio.....	57
Figura 20: Pestaña de Configuración.....	59
Figura 21: Pestaña de Estadísticas.....	60
Figura 22: Prueba N° 1.....	61
Figura 23: Prueba N° 2.....	62
Figura 24: Prueba N° 3.....	62
Figura 25: Prueba N° 4.....	63
Figura 26: Prueba N° 5.....	63
Figura 27: Prueba N° 6.....	64
Figura 28: Prueba N° 7.....	64
Figura 29: Prueba N° 8.....	65
Figura 30: Prueba N° 9.....	65

Figura 31: Prueba N° 10. .... 66

Figura 32: Unidades Remotas conectadas. .... 68

Figura 33: Aplicación de PC y Unidad Central. .... 68

Figura 34: Cartel de Entrada..... 69

# Lista de Tablas

Tabla 1: Prueba de procesamiento de imágenes ..... 52

Tabla 2: Análisis de costos de insumos. .... 72

# Lista de Diagramas

Diagrama 1: Bloques principales del sistema.....	20
Diagrama 2: Flujo de una UR.....	24
Diagrama 3: Flujo de la UC. ....	31
Diagrama 4: Flujo del procesamiento digital de imágenes.....	37
Diagrama 5: Flujo del SVM.....	46
Diagrama 6: Flujo del OCR. ....	51

## **Lista de Abreviaciones**

UR: Unidad Remota.

UC: Unidad Central.

OCR: Optical Character Recognition.

SVM: Support Vector Machine.

IR: Infrarrojo.

RNA: Red Neuronal Artificial.

MLP: Perceptrón de múltiples capas.

## **Dedicado a:**

Nuestros padres, hermanos, familiares, amigos y a la memoria del ing. Francisco Rubinelli.



## Capítulo 1. **Introducción**

A nivel nacional la tecnología aplicada a las playas de estacionamiento es mínima, utilizando en exceso el recurso humano para realizar las actividades de control de los puestos de estacionamiento y de cobro del servicio, generando demoras en la circulación de los vehículos.

El proyecto tiene como objetivo automatizar el control de una playa de estacionamiento e innovar la tecnología que se encuentra actualmente disponible en las mismas. El sistema consiste en la detección de vehículos y mediante procesamiento digital de imágenes la lectura de su patente por medio de cámaras para determinar la hora de ingreso a la playa de estacionamiento, guardar esa información en una base de datos para después en su salida calcular el tiempo de uso del lugar del estacionamiento y proceder a la facturación del mismo. Además, lleva un control de los lugares disponibles y ocupados dentro de la playa y agiliza el ingreso y egreso de vehículos. Reduciendo la cantidad de operarios que realizan estas tareas. Para llevar a cabo este proyecto se tomó como referencia una playa de 60 lugares, con una entrada y una salida, con la cabina de cobro entre medio de ambas.

Una vez ingresado, el vehículo se encontrará con un cartel electrónico, que mediante indicadores luminosos mostrará cuales son los lugares disponibles, ocupados y reservados mediante un código de colores, para que el conductor tenga una orientación rápida de hacia dónde debe dirigirse para estacionar. Cada lugar está señalizado con un semáforo de dos estados, verde y rojo, siendo el verde disponible y el rojo ocupado o reservado. El estado del semáforo es controlado por un sensor que detecta la presencia o no del vehículo y por el software que dispondrá el operario.

La aplicación de usuario permite observar en la computadora el estado actual de ocupación de los lugares, reservar lugares para alquiler mensual y poder establecer el precio de la tarifa. Al momento de salir un vehículo, el operario verá en un cuadro de información los datos del mismo, hora de ingreso y egreso, el tiempo que estuvo dentro de las instalaciones y el monto total a cobrar.

Desde la calle se podrá observar un cartel que indicará cuando no existan más lugares disponibles dentro del estacionamiento, el cual se encenderá automáticamente al momento de ocuparse todos los lugares.

Con el sistema propuesto en este proyecto y considerando la falta de uso de tecnología en la mayoría de las playas de estacionamiento de la región, se logrará una modernización de las mismas, agilizar el flujo de entrada y salida de los estacionamientos logrando evitar una congestión de vehículos en la calle durante las horas pico, llevar un registro de uso del estacionamiento, para realizar un balance económico y poder contrastar lo registrado por el sistema con lo efectivamente cobrado, además, se lograría una ventaja económica ya que se reduciría la cantidad de empleados para llevar a cabo estas tareas y minimizar el gasto en personal.

Actualmente en la ciudad de Paraná las playas de estacionamiento no cuentan con la tecnología propuesta. Ninguna posee un sistema de señalización de lugar y el mayor grado de avance encontrado fue la implementación de una máquina que emite un ticket con el horario de entrada e indica el lugar en el que se debe estacionar el vehículo, pero, la gran mayoría realiza todas las tareas de forma manual y solo cuentan con un sistema de PC para llevar el control.

La tecnología de detección de patentes se está comenzando a usar en el país para el control de tráfico en las grandes ciudades, y la señalización de lugares de estacionamiento es casi inexistente, por lo que este proyecto proporciona un avance en lo que respecta a las playas de estacionamiento, imitando lo que sucede en países vecinos y genera una nueva área de aplicación para el creciente uso del procesamiento de imágenes para el control vehicular.

## Capítulo 2. Desarrollo

### 2.1 Hardware

#### 2.1.1 Diagrama de Bloques

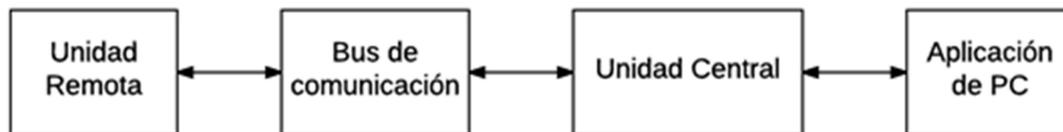


Diagrama 1: Bloques principales del sistema

Las Unidades Remotas son las encargadas del control y sensado de cada lugar dentro de la playa, para este proyecto se planteó de referencia una con 60 lugares, por lo que existirán un igual número de estas unidades. La Unidad Central es la interfaz entre la aplicación de PC y las Unidades Remotas. Por último el operario contará con una aplicación donde podrá ir observando los estados en los que se encuentran los lugares, los vehículos que ingresan y egresan, el monto a cobrar y permitiéndole realizar la reserva de lugares.

#### 2.1.2 Unidad Remota (UR)

Esta unidad está compuesta principalmente por tres elementos, uno de detección, uno de control y uno de accionamiento.

El elemento de control es el microcontrolador PIC16F1826 de Microchip. El mismo debe controlar la placa del relé, el sensor y comunicarse con la Unidad Central (UC) mediante el puerto de comunicación. Es un dispositivo económico y cumple con las exigencias mínimas para llevar a cabo las tareas anteriores. Cada Unidad Remota (UR) tiene una dirección, correspondiente al lugar del estacionamiento, la misma se encuentra grabada en el código del microcontrolador. Este, además, controla el sensor utilizado para detectar la presencia de vehículos, enviándole un pulso de activación y midiendo su respuesta y se encarga de manejar los estados del relé, a partir de la detección del sensor o por una orden de reserva de la UC.

Como elemento de detección del vehículo se utilizó el sensor ultrasónico HC-SR04 mostrado en la Figura 1 que posee un rango de detección de hasta 4[m] y gran precisión. Este estará ubicado a 3[m] de altura en el centro del lugar de estacionamiento. Necesita de un pulso de al menos 10[us] en alto en su pin de “Trig” para activarlo y responde con un pulso en alto en su pin “Echo”, cuyo ancho es proporcional a la distancia a la que rebota la señal ultrasónica que emite, si esta es menor a los 3[m] se interpreta que un vehículo se ha estacionado.

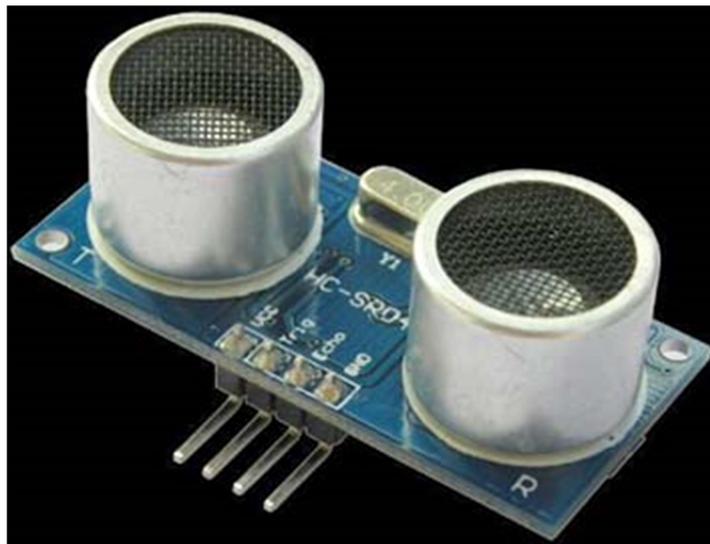


Figura 1: Sensor ultrasónico HC-SR04.

El elemento de accionamiento es un relé, cuyo contacto normalmente cerrado se utiliza para indicar que el lugar está disponible y al activarse se produce el cambio de estado.

El circuito de activación se observa en la Figura 1, en la misma se aprecia el diodo de marcha libre D1 en paralelo con la bobina del relé, la cual se alimenta al activarse el transistor Q1, siendo este último comandado por el microcontrolador. R1 establece la adecuada corriente de base y consecuentemente la de colector que fluye por la bobina.

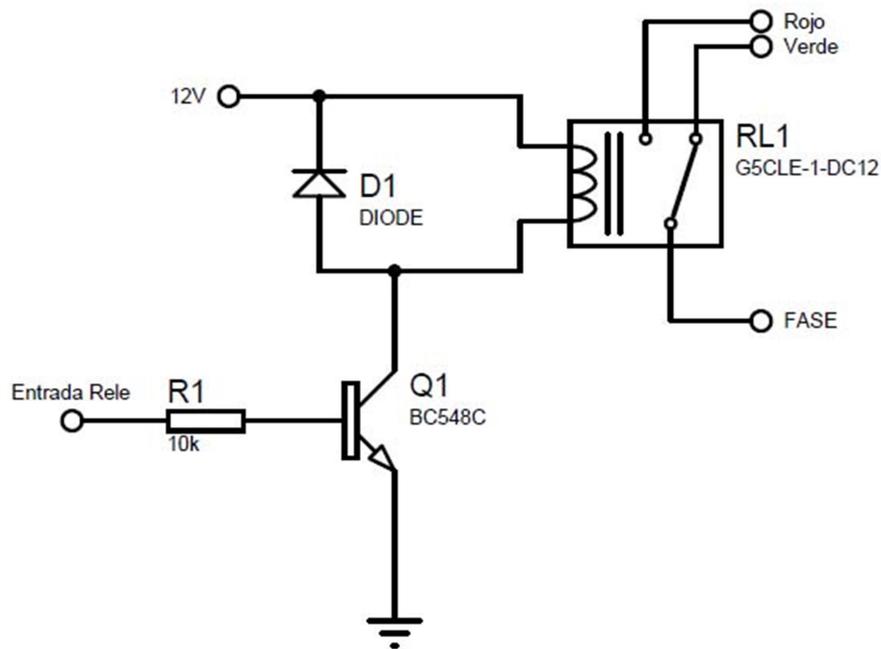


Figura 2: Circuito del relé.

Los semáforos tienen dos estados posibles, libre y ocupado, para indicar que el lugar se encuentra libre se utiliza una luz de color verde para los estacionamientos comunes y azul para los lugares reservados para personas con discapacidades, para indicar que el lugar se encuentra ocupado se utiliza una luz de color rojo en ambos casos. Para estos indicadores luminosos se utilizan lámparas tipo gotas LED, las cuales tienen buena visibilidad y son fáciles de reemplazar en caso de avería, ya que se colocan en los portalámparas convencionales E27.



Figura 3: Lámparas tipo gota LED.

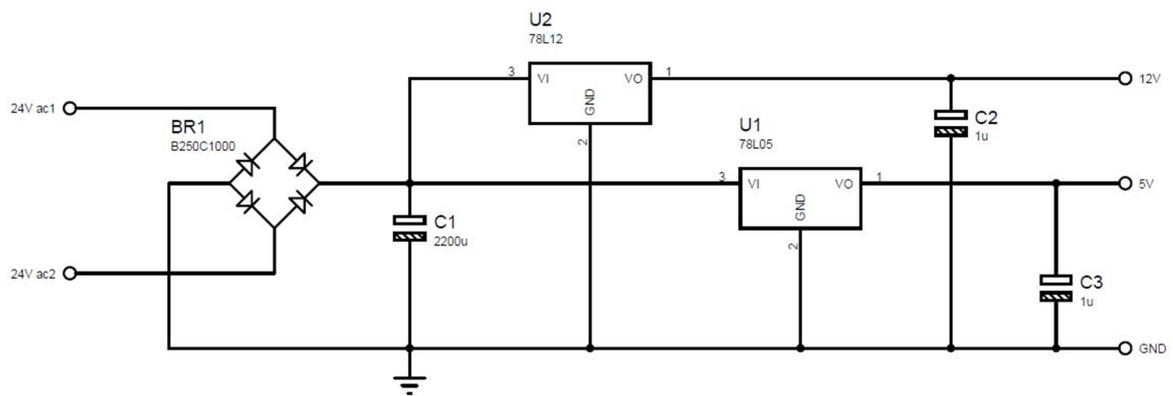
Fuente de alimentación:

Figura 4: Fuente de alimentación de una UR.

A cada UR le llegan dos cables con una tensión alterna de 24[V] provenientes de la salida del transformador de la UC. Esta tensión se aplica al puente rectificador BR1. Luego de pasar por el capacitor de ripple C1 se obtiene una tensión continua que es utilizada como referencia para los reguladores de tensión, 78L05 y 78L12, ambos con un encapsulado TO-92 y una corriente máxima de 100[mA], que es suficiente para cubrir el consumo de la UR. Cada regulador tiene un capacitor de 1[µF] a la salida para filtrar ruidos.

Desarrollo de firmware:

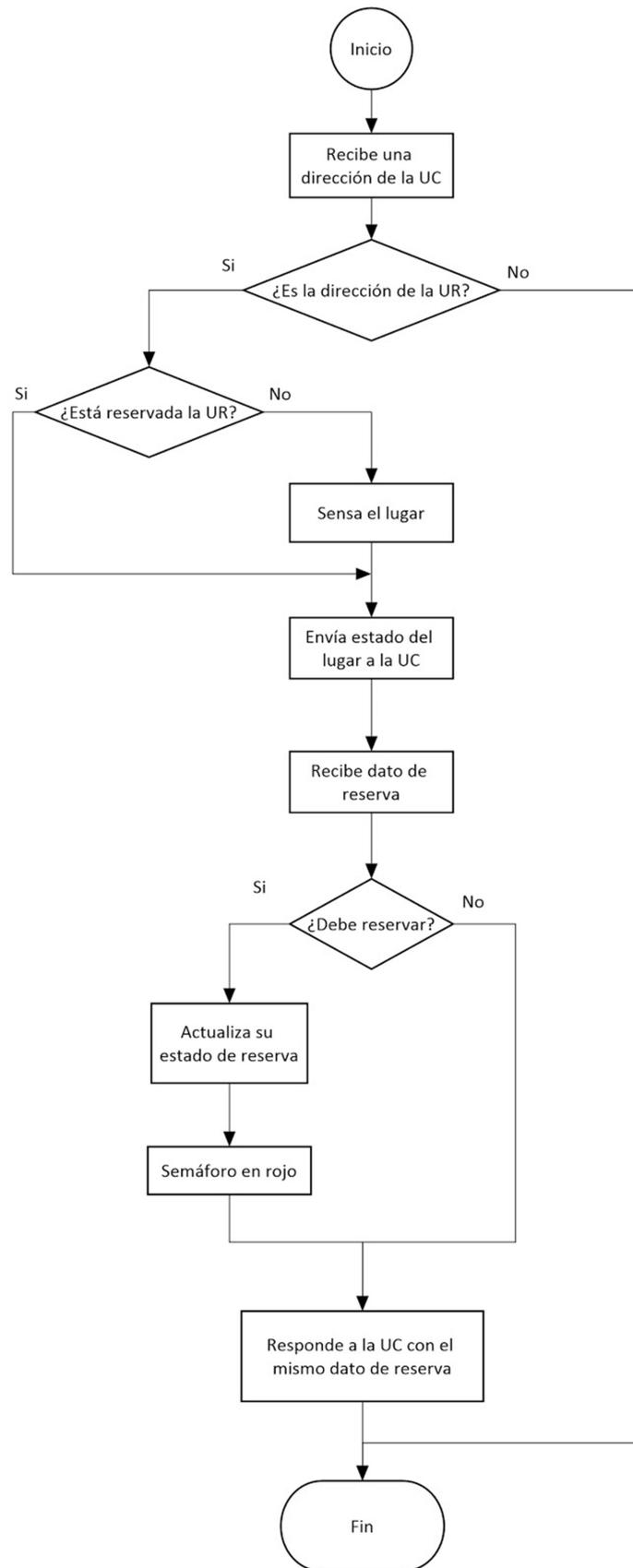


Diagrama 2: Flujo de una UR.

Todas las unidades reciben por el BUS de comunicaciones una dirección proveniente de la UC, sin realizar actividad hasta que la recibida coincida con la propia de su unidad. Una vez que el dato coincide, la UR procede a sensar el lugar para determinar si hay o no estacionado un vehículo e informa a la UC. En caso de que dicha posición se encuentre reservada, no se produce el sensado y se informa a la central que el estado del semáforo se encuentra en rojo. Seguidamente la UR recibe un dato que corresponde a si se debe reservar la posición (por alquiler mensual o por algún otro motivo) o no, luego actualiza su configuración para saber si debe sensar el lugar o no en el próximo polling. Para finalizar la comunicación se responde a la UC el dato de reserva.

Para evitar errores en el sensado o cambios involuntarios del semáforo mientras el vehículo se estaciona o si una persona cruza por debajo del sensor, la UR debe sensar 30 veces consecutivas lo mismo para cambiar de estado, ya sea para pasar de libre a ocupado o viceversa. Debido a que se produce un solo sensado por polling, el cambio de estado se realizará luego de comunicarse 30 veces con la UC, lo que lleva un tiempo aproximado de 2[s].

En el prototipo se configuró el sensor para detectar objetos a una distancia máxima de 1[m] para realizar pruebas y demostraciones. Si se detecta un objeto a esta distancia o a una menor, la UR interpreta que hay un vehículo estacionado. En el caso de implementarse realmente, hay que tener en cuenta las superficies inclinadas de los vehículos (techo, parabrisas, luneta trasera, etc.), que pueden producir que el pulso sónico no regrese al receptor del sensor. Por esta razón, se debe tomar una franja para la detección, es decir, si la distancia sensada está entre 2,90 y 3,10[m] el semáforo se debe poner en verde. De lo contrario debe ponerse en rojo. Se logra así, al colocar el sensor a una altura de 3[m] aproximadamente, detectar el piso cuando no hay un auto estacionado, y de estacionarse uno, la distancia sensada será menor a la franja establecida o bien superior si el pulso rebota en una superficie inclinada.

Las unidades remotas no tienen ningún código de detección de errores de comunicación, esto se realiza solo en la Unidad Central.

### 2.1.3 Problemas y soluciones implementadas

La UC haría las veces de “Maestro” y cada UR de “Esclavo” pero el puerto UART está pensado para una comunicación punto a punto y su lógica de funcionamiento, junto con las limitaciones físicas como la distancia y el voltaje de sus niveles lógicos, hacían imposible su uso para más de dos estaciones. Es por ello que se decidió el uso del BUS de comunicaciones RS-485 cuyas características nos permiten un gran número de estaciones y distancias grandes (hasta 1200[m]) con una alta tasa de transmisión.

Para poder realizar lo anterior, se buscó un microcontrolador que cuente con estas características y sea de bajo costo. En primera medida se pensó en el PIC16F628A el cual cuenta con módulo USART y un módulo CCP, el primero encargado del manejo de la comunicación serie y el segundo de la medición del pulso enviado por el sensor ultrasónico, posteriormente se realizaron pruebas con un microcontrolador PIC16F1826 que pertenece a la línea mejorada de los PIC16F. Se notó una mejora en la comunicación entre los dispositivos y debido a que su costo es inferior al primero, se optó por la utilización de este último.

Al momento de probar las unidades, se pensó originalmente que todos los sensores fueran activados de forma simultánea, pero, apareció el inconveniente de que estos se interferían entre si, provocando falsas detecciones que activaban los rele y generaban cambios de estado erróneos en los semáforos. Para solucionar esto, se programó la UC para que realice un polling indicándole a cada UR que debe realizar el sensado y de esta manera evitar la interferencia. Cabe destacar que el tiempo que se demora en realizar el recorrido por todas las unidades es insignificante para el funcionamiento del sistema.

Los semáforos convencionales están contruidos a partir de LEDs de colores de alto brillo. Para este proyecto, se partió de la idea de construirlos en vez de comprarlos por el alto precio y consumo que tienen. Pero el problema era que el precio de cada LED de alto brillo es elevado y la cantidad que se necesita sumado al tiempo necesario para fabricar las placas donde van montados obligó a descartar la idea. Para solucionar este inconveniente se decidió el uso de lámparas tipo gota LED, que tienen como ventajas poco consumo, fácil colocación y reemplazo, buena visibilidad y bajo precio.

En un principio se pensó en la implementación de una fuente capacitiva [4] en cada UR para evitar un tendido adicional de cable de alimentación, aprovechando los cables de

línea necesarios para energizar las lámparas LED. Al momento de implementar este tipo de fuente se encontró que, si bien entregaba los valores de tensiones de salida correctas, cuando se le quitaba el fusible de la fase que alimentaba a la misma, a través de los cables de comunicación que conectan las distintas unidades se producía una tensión alterna de 220[V] a la entrada de la fuente desconectada, lo que produjo daños en varios componentes de los circuitos. Para solucionar esto se cambió el tipo de fuente por una convencional compuesta por reguladores de tensión, siendo todas alimentadas por un mismo transformador de 24[V] de salida.

#### **2.1.4 BUS de comunicaciones**

Cada UR debe comunicarse con la UC de manera bidireccional. Debido a que son varios dispositivos los que se encuentran conectados al BUS de comunicaciones y teniendo en cuenta que alguno de estos están a distancias lejanas de la UC, se implementó un BUS de comunicaciones con el protocolo RS-485 el cual permite conectar varios dispositivos con una distancia de hasta 1200[m]. Para adaptar los niveles de tensión se utilizó el integrado MAX487 que además permite controlar si la unidad debe leer el BUS o escribir en él. Según la norma se aconseja el uso de un par trenzado de cables para aumentar la inmunidad al ruido.

Para realizar la comunicación se implementó un protocolo Maestro/Esclavo, en el cual la UC envía la dirección de la UR con la que quiere comunicarse, ésta le responde con el estado del lugar que está sensando, finalmente la UC le envía un dato indicando si tiene que reservar o no el lugar (en caso de que un cliente decida pagar el alquiler de forma mensual y la playa decida asignarle un lugar fijo) y debe recibir un dato de confirmación de parte de la UR. Esta comunicación es del tipo half-duplex dado que no puede haber dos unidades transmitiendo al mismo tiempo, en la Figura 5 se observa una conexión típica.

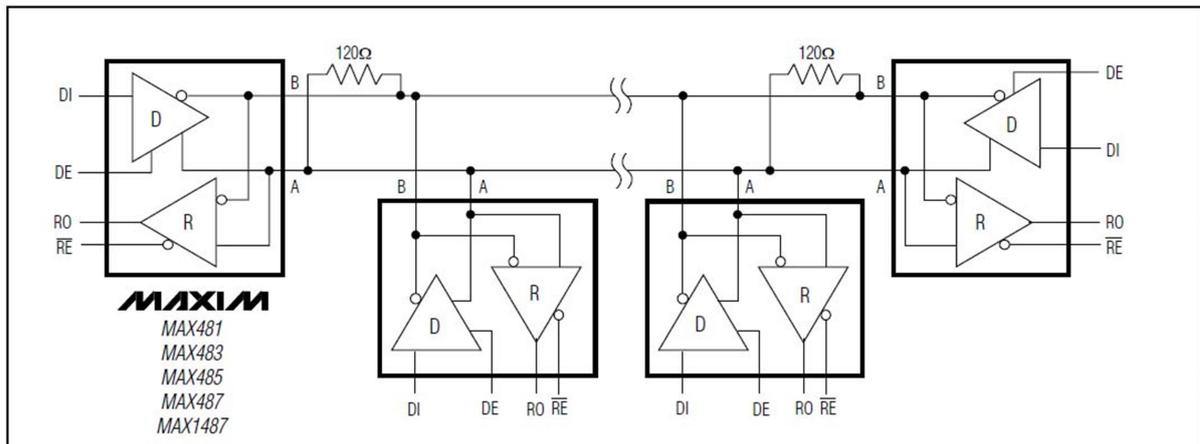


Figura 5: Conexión típica Half-Duplex RS-485.

En la Figura 6 se puede observar la conexión del integrado, donde RX y TX se conectan a los pines del puerto UART del microcontrolador y la señal de CONTROL es la encargada de habilitar la lectura o escritura en el BUS, R1 es una resistencia de pull-down para poder trabajar de manera correcta con los estados lógicos del PIC, R2 es la resistencia de terminación de línea utilizada para que no haya reflexión de onda, según indicaciones de la hoja de dato su valor es de 120[Ω] y deben estar ubicadas al principio y final de la línea de comunicación, en este caso estarán ubicadas en la UC y en la última de las URs.

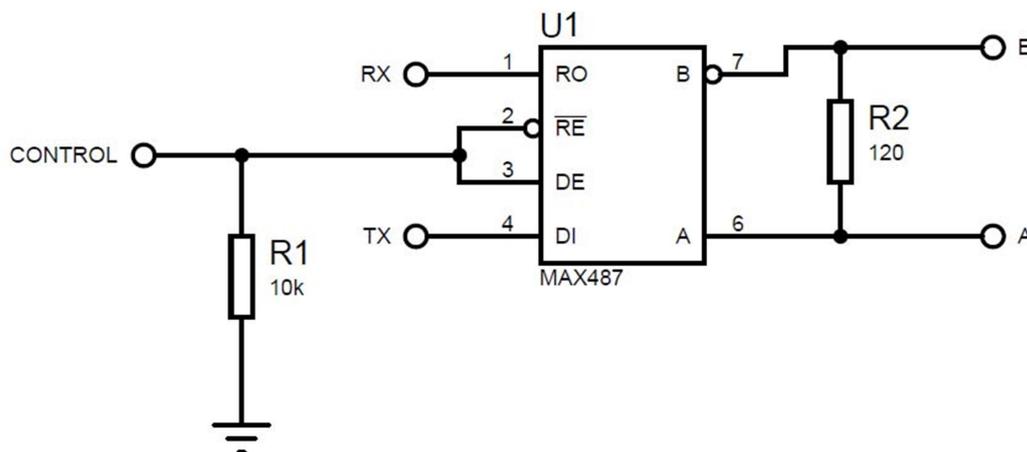


Figura 6: Transceptor MAX487.

### **2.1.5 Problemas y soluciones implementadas**

Al comenzar las pruebas, la señal de CONTROL era aplicada sin la resistencia de pull-down, según lo indicado en la hoja de datos, y cuando el microcontrolador ponía en estado bajo el pin que genera esta señal, el MAX487 interpretaba un estado erróneo produciendo que más de una unidad quiera escribir en el BUS al mismo tiempo. Al conectar la resistencia de pull down se eliminó este error, logrando que un solo dispositivo esté transmitiendo, además, se produjo una reducción significativa en el consumo de cada UR.

En las pruebas realizadas se presentaron problemas de comunicación entre las URs y la UC, como falsos cambios de estado y reservas equivocadas, las cuales se debieron a problemas con el par trenzado, que si bien aparentaban continuidad, en el proceso de comunicación presentaban fallas. Al reemplazarlos por otros del mismo tipo se solucionó el inconveniente.

### **2.1.6 Unidad Central (UC)**

Esta unidad está compuesta por un elemento de procesamiento y dos puertos de comunicaciones (UART y USB).

El elemento de procesamiento es el microcontrolador PIC18F14K50 de Microchip. El mismo debe realizar la comunicación con la aplicación de PC a través del puerto USB y con las URs a través del BUS de comunicación. Es un dispositivo económico y cumple con las exigencias mínimas para llevar a cabo las tareas anteriores.

En lo que respecta a la comunicación con las URs, la misma se realiza por polling, siendo la UC la que envía a través del BUS como primer dato la dirección de la UR con la que desea comunicarse, esta le responde con su estado de semáforo, y la UC procede a enviarle el dato de reserva, a lo que le es respondido con un dato de confirmación. Luego se repite el proceso anterior con la dirección de la próxima UR.

La comunicación con la PC consta de dos etapas, la primera es cuando detecta un cambio de estado en alguna de las URs, por lo que procede a comunicarlo para que la aplicación actualice la lista de lugares mostrada en pantalla, la cual indica el estado de ocupación de los mismos.

La segunda etapa corresponde a cuando el operario indique la reserva de un determinado lugar. La UC debe recibir este pedido desde la aplicación y actualizar el estado de las URs en el siguiente polling.

La comunicación con las URs se realiza en forma continua, mientras que la comunicación hacia la PC se produce únicamente cuando se detecta un cambio de estado en alguna de las URs o cuando la aplicación le envía una orden de reserva, interrumpiendo el proceso de polling.

#### Fuente de alimentación:

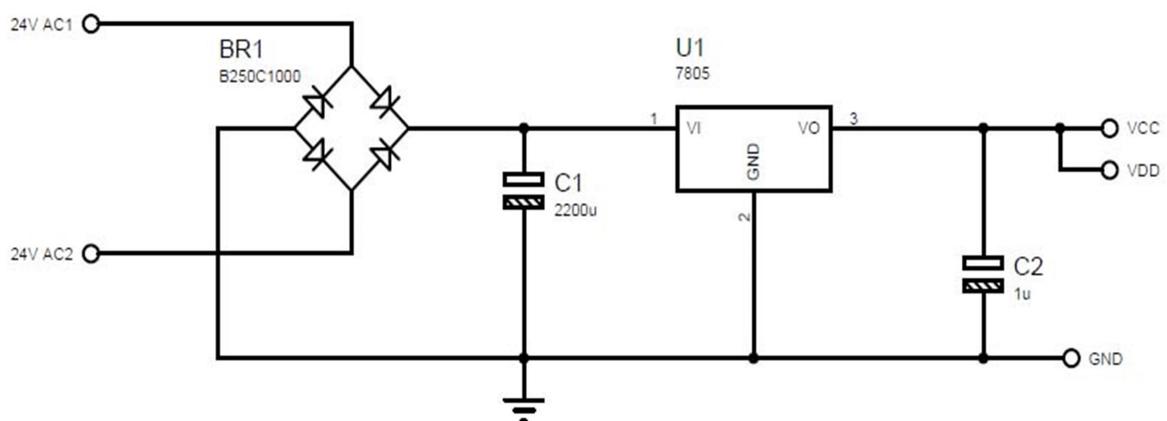


Figura 7: Fuente de alimentación de la UC.

Es similar a la fuente de alimentación de una UR con la diferencia de que solo se necesita una tensión continua de 5[V] para el funcionamiento de la UC dado que no posee ningún elemento de control (relé de 12[V]). La salida de 24[V] del transformador se aplica al puente rectificador BR1 y luego del capacitor de ripple C1 se obtiene una tensión continua de referencia para el regulador de tensión 7805, el cual tiene conectado el capacitor C2 a su salida para filtrar ruidos.

Desarrollo de firmware:

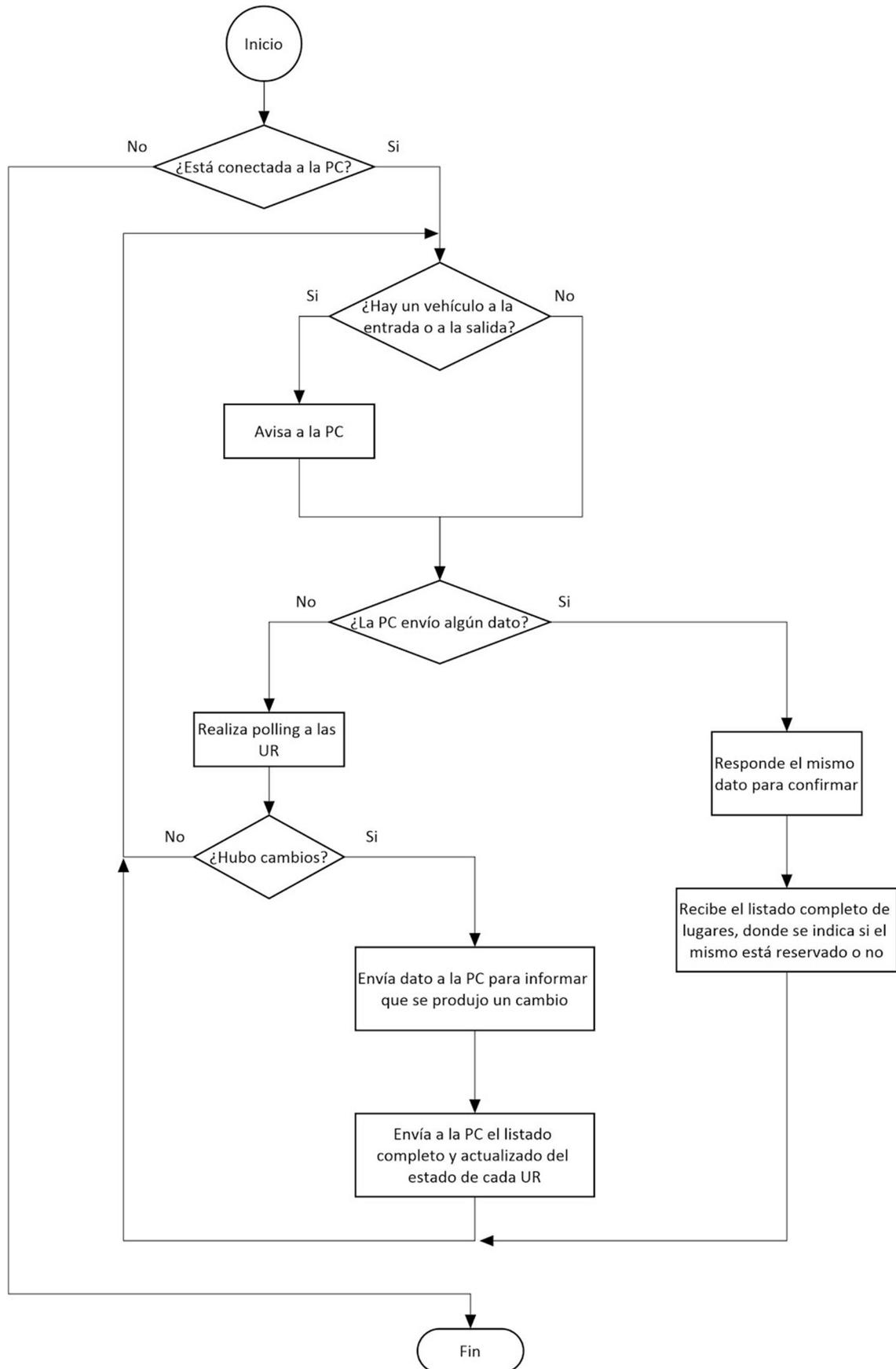


Diagrama 3: Flujo de la UC.

La Unidad Central es la encargada de realizar la comunicación entre la aplicación de PC y las URs y de llevar a cabo el polling con estas últimas para ir actualizando el estado de cada lugar dentro de la playa.

El sistema solo funcionará si se encuentra conectado a la PC, de caso contrario permanecerá sin realizar actividades. Si se produjo la conexión la UC es la encargada de chequear los sensores ubicados a la entrada y salida del estacionamiento, los cuales se activan cuando un vehículo pasa sobre ellos, en caso de que alguno se active se le informa a la PC para que esta proceda a la captura e identificación de la patente.

Existen dos tipos de comunicación posible y ambas son bidireccionales, una de ellas es con las distintas URs mediante un BUS de comunicación y mediante el protocolo RS-485, la otra es con la computadora mediante USB. En el primero de los casos se estableció una comunicación del tipo maestro/esclavo donde la UC cumple con el rol de maestro interrogando a cada UR sobre el estado en el que se encuentran e indicándoles a las mismas si deben permanecer reservadas o no. Se utilizó el siguiente protocolo de comunicación:

1. El maestro envía dirección de un esclavo mientras estos escuchan.
2. El esclavo que tiene la dirección enviada anteriormente responde con el estado de su semáforo, mientras el resto de las unidades solo escuchan.
3. El maestro envía el dato de reserva correspondiente a ese esclavo.
4. El esclavo que mantiene la conversación con el maestro retiene este dato y actualiza su estado, mientras que las demás unidades lo descartan.
5. El esclavo responde con el mismo dato de reserva para finalizar la comunicación.
6. El maestro inicia la misma secuencia con el esclavo que tiene la dirección siguiente.

La comunicación entre la UC y las URs se produce en el orden de los microsegundos, en caso de falla de respuesta de alguna de las unidades, pasado un tiempo, del orden de los milisegundos, la central procede a comunicarse con el siguiente esclavo. No se realiza detección de errores de comunicación en este proceso debido a que, como se mencionó anteriormente esta se produce de manera muy rápida y se ejecuta varias veces por segundo, de modo que si hubo un error, el mismo se corregirá en el siguiente polling y no será percibido.

Con la PC no se implementó el sistema maestro/esclavo, sino que la comunicación se produce cuando ocurre algún evento en cualquiera de las dos partes. Si se produjo el cambio de estado en alguna de las URs, una vez finalizado el polling, la UC enviará un

dato a la PC indicándole que va a comenzar una conversación para actualizar el estado de los lugares, para este caso se utilizó el siguiente protocolo de comunicación:

1. La UC envía un dato para iniciar la comunicación.
2. La PC responde el mismo dato para indicar que está lista para la comunicación.
3. La UC envía el estado del primer lugar de la playa.
4. La PC responde con el mismo dato.
5. La UC envía el estado del siguiente lugar.
6. La PC responde con el mismo dato.

Una vez que se envían todos los lugares se termina la comunicación.

En esta comunicación la UC compara que el dato recibido por parte de la PC sea el mismo que se envió, en caso de que se haya producido un error en la comunicación, se vuelve a iniciar la comunicación y se repiten los pasos mencionados anteriormente, si la comunicación fue correcta, la UC pasa a realizar el polling con las distintas URs.

En caso de que el operario decida reservar un determinado lugar dentro de la playa, es la PC quien inicia la comunicación, enviando un dato a la UC, una vez que esta última se encuentra en condiciones de establecer la conversación, es decir, termina de comunicarse con las URs, responde a la PC indicándole que está disponible y suspende el polling hasta terminar la comunicación. Para este caso se utilizó el siguiente protocolo:

1. La PC envía un dato para iniciar la comunicación.
2. La UC responde el mismo dato para indicar que está lista para la comunicación.
3. La PC envía el dato de “reserva” o “no reserva” del primer lugar de la playa.
4. La UC responde con el mismo dato.
5. La PC envía el dato del siguiente lugar.
6. La UC responde con el mismo dato.

Una vez que se envían todos los lugares se termina la comunicación y la UC comienza a realizar el polling con las URs.

En este caso es la PC quien debe encargarse de controlar que no se hayan producido errores y en caso de que hayan sucedido volver a iniciar la comunicación.

### **2.1.7 Problemas y soluciones implementadas**

El problema principal que se tuvo fue poder trabajar con los dos tipos de comunicaciones. Al principio se pretendió que las interrupciones de la UART y del USB estén activas y sea el microcontrolador el encargado de atender una u otra

dependiendo de quién se esté comunicando. Al no lograr que el PIC respete los niveles de prioridad asignados a las interrupciones se optó por deshabilitar una mientras se trata con la otra.

### 2.1.8 Cartel de entrada

El cartel se comporta como una UR que siempre escucha, es decir, le llegan todas las direcciones de las URs y debe indicar los estados de los 6 lugares implementados, para ello utiliza dos juegos de 6 LEDs, uno verde (azul para el lugar de personas con discapacidad) y uno rojo para representar el estado libre y ocupado/reservado respectivamente, de cada lugar.

Es poco eficiente utilizar cada pin de un microcontrolador para controlar el estado de los LEDs del cartel que representan los distintos lugares del estacionamiento, debido a que no se podría utilizar un único microcontrolador para cubrir la totalidad de espacios disponibles en una playa.

Es por ello que se utilizó junto con el PIC 18F14K50, el decodificador de BCD a decimal HEF4028B para crear un sistema de multiplexación que permitiese aumentar la cantidad de lugares que debe mostrar el cartel sin tener que aumentar el número de integrados.

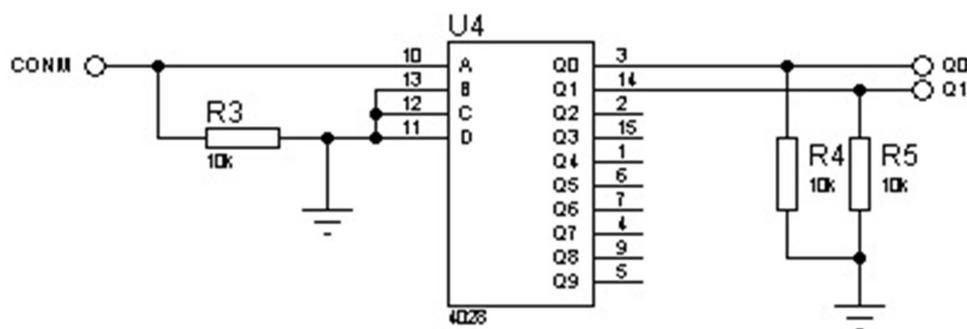


Figura 8: Decodificador de HEF4028B.

Como se observa en la Figura 8, el decodificador tiene 10 salidas y 4 entradas, para el caso del prototipo solo se utilizó dos salidas (Q0 y Q1) controladas por la entrada A. El cartel se construyó como una matriz de 3 columnas y 2 filas, dentro de cada columna hay un juego de LEDs verdes y uno de LEDs rojos, cada juego es controlado por un puerto del microcontrolador y cada fila por una salida del decodificador. La salida Q0 se activa cuando la entrada A está en bajo y la Q1 cuando A está en alto. Conmutando

rápidamente entre las salidas se logra el efecto visual de ver todas las columnas activas (ya sea en verde o rojo).

De implementar el prototipo para 60 se utilizarían todas las salidas del decodificador, 4 pines del microcontrolador para controlar sus entradas y 12 pines más para las columnas (2 pines por columna, uno para los LEDs rojos y otro para los verdes). Utilizando en total 16 pines de entrada/salida del microcontrolador 18F14K50. De esta forma se crearía una matriz de 6 columnas y 10 filas para representar el cartel de 60 lugares.

El juego de LEDs verdes y el de rojos se montan sobre una pequeña placa sobre la que se disponen de a pares intercalados, es decir, 2 rojos, 2 verdes y así, conformando una matriz de 2 columnas (LEDs del mismo color) y 6 filas como se observa en la Figura 9.

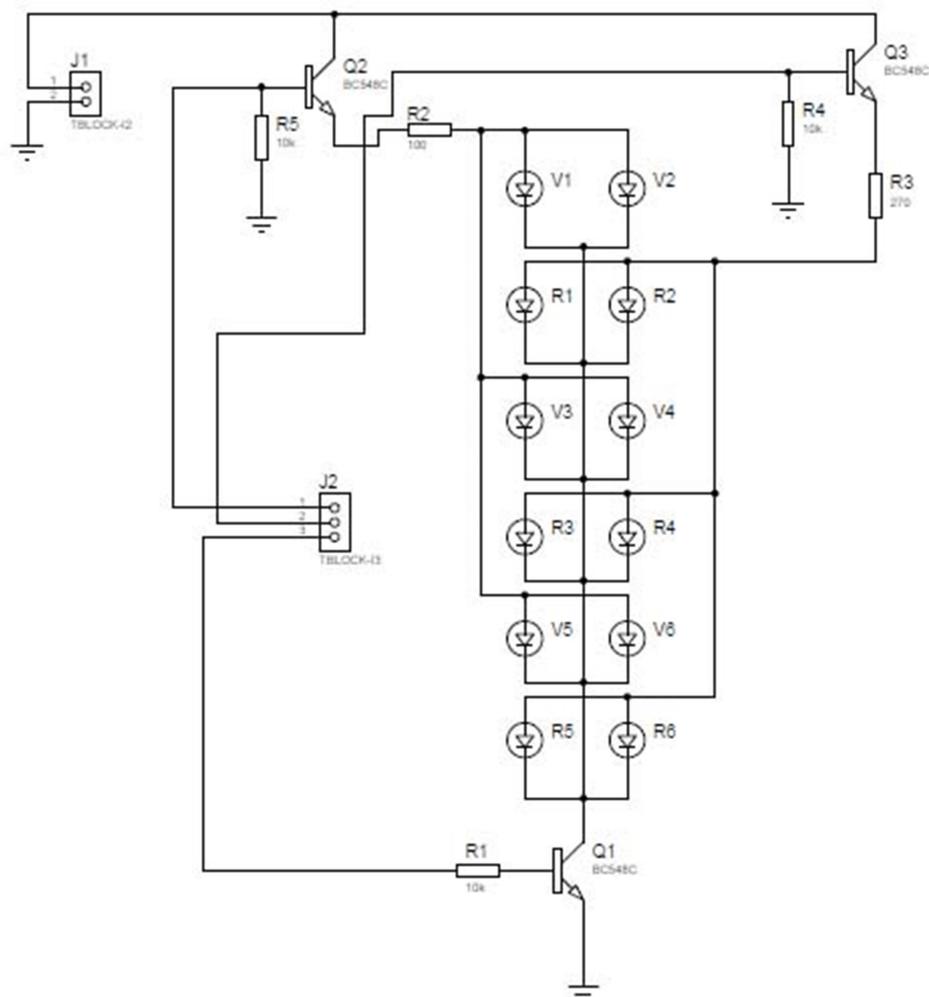


Figura 9: Circuito esquemático del bloque de LEDs.

Cada juego es controlado por un transistor BC548 y sus resistencias de polarización. El juego de rojos (LEDs R1 a R6) se controla con Q2 y las resistencias R2 y R5. Mientras que el de verdes (V1 a V6) con Q3, R3 y R4. El conector J1 proporciona la tensión de colector de ambos transistores, la cual es de 5[V]. Las bases de estos transistores son controladas por los pines del microcontrolador que se conectan a las posiciones 1 y 2 del conector J2. La posición 3 de este conector es la salida Q0 o Q1 del decodificador, dependiendo en qué fila se encuentre el bloque de LEDs. Si la salida del decodificador está en alto se habilita la masa para los LEDs permitiendo que se enciendan los que correspondan al transistor activado arriba (Q2 o Q3).

La fuente de alimentación del cartel de entrada es la misma que la de la UC utilizando un regulador de tensión 7805 con un encapsulado TO-220 que permite hasta 1[A] de corriente de salida dado que debe alimentar a todas las placas de LEDs (6 placas con 6 LEDs cada una).

## 2.2 Procesamiento Digital de Imágenes

### 2.2.1 Diagrama de bloques

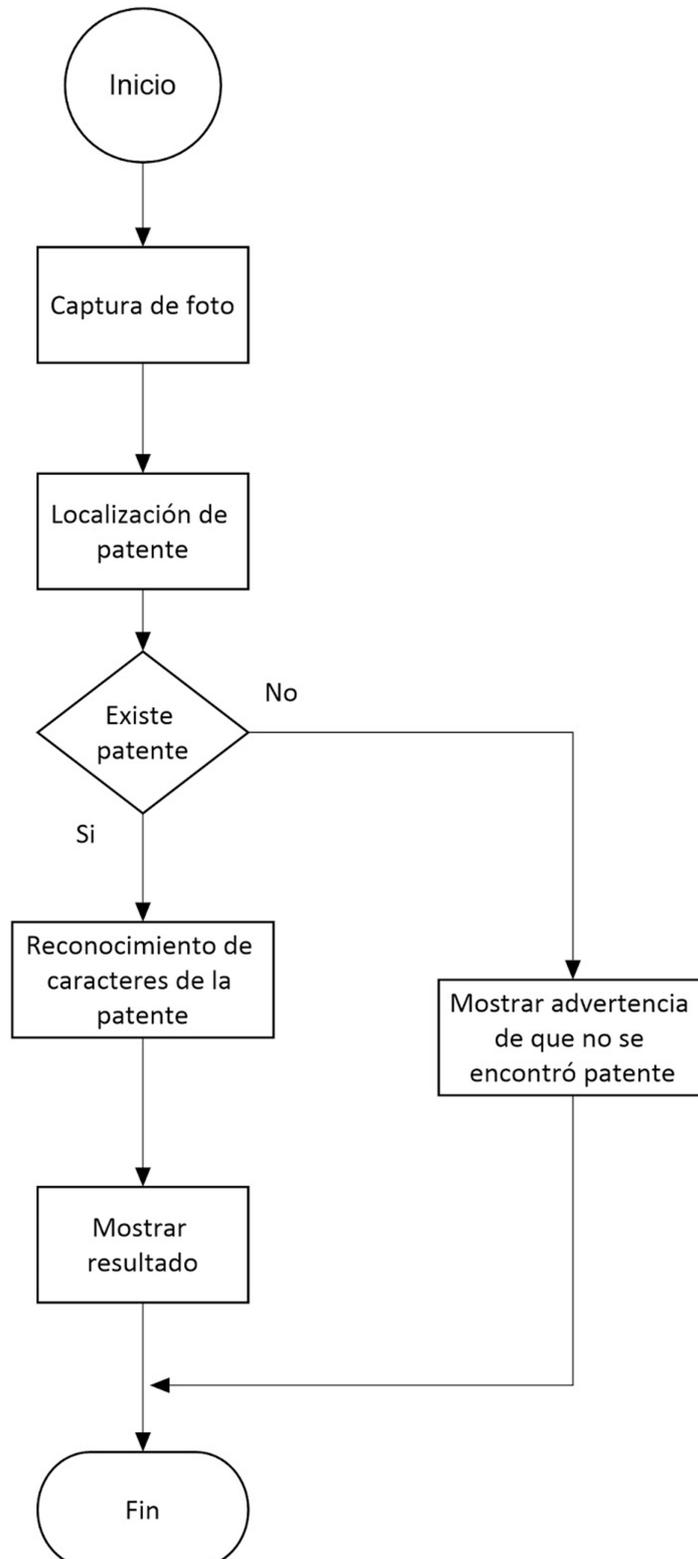


Diagrama 4: Flujo del procesamiento digital de imágenes.

Para el reconocimiento y lectura de las patentes de los vehículos lo primero que se debe hacer es capturar una foto del mismo, la cual se consigue mediante una cámara IP conectada a la misma red de la PC, a partir de esta imagen la aplicación busca las regiones que contengan posibles patentes para después detectar las letras y números que poseen. [1]

### **2.2.2 OpenCV**

OpenCV (Open Source Computer Vision Library) es una librería de visión computacional de fuente abierta que se publica bajo una licencia BSD y por lo tanto es gratuito para uso académico y comercial. Tiene interfaces C++, C, Python y Java y soporta Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Escrito en C / C++ optimizado, la biblioteca puede aprovechar el procesamiento multi-core.

Adoptada en todo el mundo, OpenCV tiene más de 47 mil personas de la comunidad de usuarios y el número estimado de descargas de más de 14 millones. Los usos van desde el arte interactivo hasta la inspección de minas, la realización de mapas en la web o la robótica avanzada.

En el caso de este proyecto, se la emplea para llevar a cabo la detección de las patentes y el reconocimiento de los caracteres de la misma. Para ello se utilizó la versión 2.4.3.13 de la librería y el entorno de desarrollo integrado (IDE, por sus siglas en inglés) utilizado para desarrollar la aplicación de PC es el Microsoft Visual Studio 2017 para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, etc.

### **2.2.3 Detección de patentes**

El objetivo de esta etapa es la detección de las distintas patentes que circulan actualmente en Argentina y el reconocimiento de los caracteres de las mismas.

Las patentes vigentes son la implementada desde 1995 y la del Mercosur desde 2016. Ambas poseen distintas dimensiones, colores, formato y tipografía de caracteres. En las siguientes figuras se observan ejemplos de las patentes, formato de 1995 (Figura 10) y del Mercosur (Figura 11).



Figura 10: Patente argentina desde 1995 a 2016.



Figura 11: Patente del Mercosur desde 2016 hasta la actualidad.

### Introducción a la Detección Automática de Patentes [1]

La detección automática de patentes es un método de vigilancia que utiliza Reconocimiento de Caracteres (OCR, Optical Character Recognition) y otros métodos tales como segmentaciones y detección para leer la patente del vehículo.

Los mejores resultados se pueden obtener con una cámara infrarroja (IR) porque los pasos de segmentación para la detección y la segmentación OCR son fáciles, limpios y minimizan los errores. Con una cámara IR se puede recuperar sólo la luz infrarroja y se tiene una imagen de muy alta calidad para segmentar y posteriormente detectar, y reconocer el número de patente, siendo independiente de cualquier entorno luminoso. En este proyecto se usa una cámara IP configurada para trabajar en modo infrarrojo.

La detección automática de patentes se divide en dos etapas principales: detección de la placa de la patente y reconocimiento de sus caracteres. La detección de placas tiene el propósito de detectar la ubicación de la placa en todo el marco de la cámara. Cuando se detecta una placa en una imagen, el segmento de placa se usa para el paso del reconocimiento de caracteres con el algoritmo de OCR.

Tanto para detectar la patente como para reconocer sus caracteres, se trabaja con los patrones que definen a la patente primero y los que definen a cada carácter después. Es por ello que es necesario definir tres pasos que se utilizan comúnmente en algoritmos de reconocimiento de patrones:

1. Segmentación: Detecta y remueve cada región de interés en la imagen. En este caso cualquier región que pueda contener una patente.
2. Extracción de características: Extrae de cada región un conjunto de características para su análisis.
3. Clasificación: Extrae cada carácter de la placa de reconocimiento o clasifica cada región de la imagen en "patente" o "no patente" en la etapa de detección de placas.

#### Detección de patente [1]

Para detectar la patente de la captura tomada por la cámara se emplean dos pasos principales: segmentación y clasificación de segmento.

En el primer paso (segmentación), se aplican diferentes filtros, operaciones morfológicas, algoritmos de contorno y validaciones para recuperar las partes de la imagen que pueden contener una patente. En el segundo paso (clasificación), se aplica un clasificador "Máquina de Vector de Soporte" (SVM, por sus siglas en inglés) en cada región de la imagen. Antes de crear la aplicación principal se entrena al SVM con imágenes de patentes y no patentes (partes del vehículo que puedan ser confundidas con la patente) en un determinado tamaño en píxeles. La aplicación toma una foto en color de vista frontal entre 2 y 4[m] de un automóvil, pudiendo trabajar en distintas condiciones de luz.

En la Figura 12 se observa todos los procesos involucrados en la detección de placas:

- Filtro Sobel.
- Operación de umbral.

- Operación morfológica cercana.
- Máscara de zona de llenado.
- Placas detectadas después del clasificador SVM.
- Caracteres detectados en la patente.



Figura 12: a) Imagen original, b) Imagen luego del filtro sobel, c) Imagen luego del filtro umbral, d) Imagen luego de operación morfológica, e) Máscara de zona de llenado f) Patente segmentada, g) Caracteres detectados.

### Segmentación [1]

La segmentación es el proceso de dividir una imagen en múltiples segmentos. Con ello se simplifica la imagen para análisis y facilita la extracción de sus características.

Una propiedad importante de la segmentación de placas es el elevado número de bordes verticales en una patente suponiendo que la imagen se tomó frontalmente, y esta no se gira y no tiene distorsión de perspectiva. Esta propiedad puede ser explotada durante la primera segmentación para eliminar las regiones.

Antes de encontrar los bordes verticales, se convierte la imagen color, capturada por la cámara, a una de escala de grises porque el color no ayuda a la detección. Se aplica un desenfoque gaussiano de 5x5 para eliminar el posible ruido generado por la cámara o por el ambiente. Si no se hiciera se podría obtener una gran cantidad de bordes que producirían una detección fallida.

Para encontrar los bordes, se utiliza un filtro Sobel y se busca la primera derivada horizontal, dado que la derivada es una función matemática que permite encontrar los bordes verticales en una imagen.

Después del filtro Sobel, se aplica un filtro de umbral (threshold) para obtener una imagen binaria con un valor umbral obtenido a través del método de Otsu. Los métodos de umbral [3] son un grupo de algoritmos cuya finalidad es separar los objetos de una imagen que interesen del resto. Normalmente los métodos del valor de umbral "binarizan" la imagen de partida, es decir se construyen dos segmentos: el fondo de la imagen y los objetos buscados. La asignación de un píxel a uno de los dos segmentos (0 y 1) se consigue comparando su nivel de gris  $g$  con un cierto valor umbral preestablecido  $t$  (en inglés *threshold*). La imagen final es muy sencilla de calcular ya que para cada píxel sólo hay que realizar una comparación numérica. La regla de cálculo correspondiente  $T_g$ :

$$T_g = \begin{cases} 0, & g \leq t \\ 1, & g > t \end{cases}$$

El punto clave es la elección del valor umbral más adecuado. El método de Otsu, llamado así en honor a Nobuyuki Otsu que lo inventó en 1979, utiliza técnicas estadísticas, para resolver el problema. En concreto, se utiliza la varianza, que es una medida de la dispersión de valores, en este caso se trata de la dispersión de los niveles de gris. El método de Otsu calcula el valor umbral de forma que la dispersión dentro de cada segmento sea lo más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre segmentos diferentes. Para ello se calcula el cociente entre ambas varianzas y se busca un valor umbral para el que este cociente sea máximo. El algoritmo de Otsu necesita una imagen de entrada de 8 bits y determina automáticamente el valor umbral óptimo. [3]

Mediante la aplicación de una operación morfológica cercana, se puede eliminar espacios en blanco entre cada línea de borde vertical, y conectar todas las regiones que tienen un alto número de bordes. En este paso se obtienen las posibles regiones que pueden contener patentes, aunque la mayoría de las regiones no contendrán placas. En las que sí haya, mediante el uso de la función *findContours* provista por OpenCV se recuperan los contornos de una imagen binaria con diferentes métodos y resultados.

Sólo se necesitan obtener los contornos externos con cualquier relación de jerarquía y resultado de aproximación poligonal.

Para cada contorno detectado, se extrae el rectángulo delimitador de área mínima. OpenCV brinda la función *minAreaRect* para esta tarea, la cual devuelve una clase de rectángulo rotado llamado *RotatedRect*. A continuación, utilizando un vector iterador sobre cada contorno, se puede obtener el rectángulo girado y hacer algunas validaciones preliminares antes de clasificar cada región.

Se realizan validaciones básicas sobre las regiones detectadas en función de su área y relación de aspecto. Sólo se considera que una región puede ser una patente si la relación de aspecto es aproximadamente  $400/130 = 3,076923$  (ancho de placa dividido por la altura de la placa en milímetros) para una patente del Mercosur y  $296/130 = 2,276923$  para una con formato de 1995, con un margen de error del 80% y un área basada en un mínimo de 15 píxeles y máximo de 250 píxeles para la altura de la placa. Estos valores se calculan dependiendo de los tamaños de imagen y la posición de la cámara.

Una vez que el proceso de segmentación está terminado y se tienen regiones válidas, se pueden recortar cada región detectada, eliminar cualquier rotación posible, recortar la región de la imagen, redimensionar la imagen e igualar la luz de las regiones de imágenes recortadas.

Después de girar la imagen, se recorta la misma y se copia una porción de la imagen de determinado ancho y alto centrada en un punto.

Las imágenes recortadas no son buenas para su uso en entrenamiento y clasificación, ya que no tienen el mismo tamaño. Además, cada imagen contiene diferentes condiciones de luz, incrementando sus diferencias relativas. Para resolver esto, se cambia el tamaño de todas las imágenes al mismo ancho y alto, y se aplica la ecualización de luz del histograma.

### Clasificación [1]

Después de preprocesar y segmentar todas las partes posibles de una imagen, se tiene que decidir si cada segmento es o no una patente. Para ello se utiliza algoritmos de Support Vector Machine (SVM).

Un SVM es un algoritmo de reconocimiento de patrones incluido en una familia de algoritmos de aprendizaje supervisado, creados originalmente para la clasificación binaria.

Este es un algoritmo de aprendizaje automático que aprende mediante el uso de datos etiquetados. Se tiene que entrenar el algoritmo con una gran cantidad de este tipo de datos. Cada conjunto de datos necesita tener una clase.

Un ejemplo clásico es un conjunto de puntos 2D que define dos clases. El SVM busca la línea óptima que diferencia cada clase:

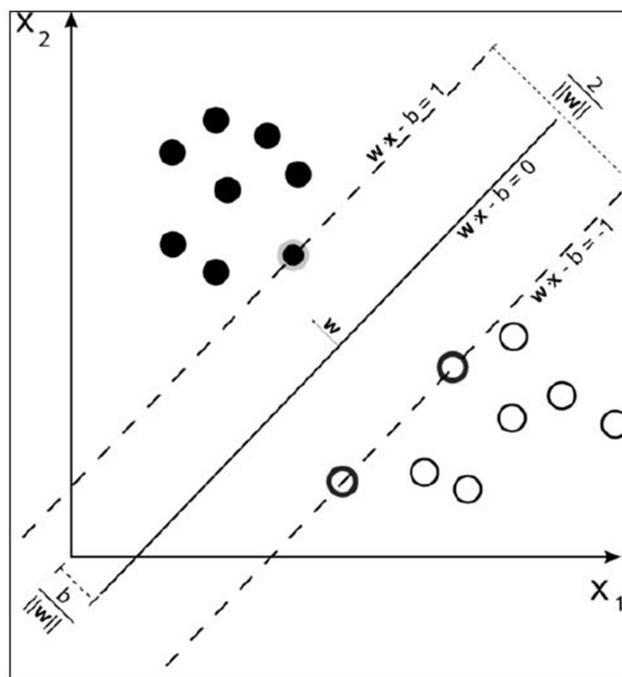


Figura 13: Ejemplo de uso de SVM.

Dado un grupo de puntos, que forma parte de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría se desconoce) pertenece a una categoría o a la otra.

La primera tarea antes de cualquier clasificación es entrenar el clasificador. Este trabajo se realiza antes de comenzar la aplicación principal y se denomina formación sin conexión. Esto requiere una gran cantidad de datos para entrenar el sistema, aunque un conjunto de datos más grande no siempre implica los mejores resultados.

Para poder obtener una cantidad de muestras significativas y lograr entrenar el sistema de forma que cumpla con los objetivos propuestos en este proyecto, se pidió autorización en la UTN FRP, para poder tomar fotos a las patentes de los vehículos que se encontraban en el estacionamiento de la institución y sumadas a las tomadas en la vía pública, se consiguieron alrededor de 200 muestras con los dos tipos de patentes vigentes, de estas se descartaron algunas imágenes debido al deterioro que presentaban algunas patentes. A las fotos que fueron seleccionadas para entrenar el sistema se les extrajo las patentes recortando las mismas en un tamaño determinado, de igual dimensiones se recortaron distintas partes del vehículo que puedan confundir al sistema, como logos, calcomanías, luces, etc. Consiguiendo un total de 112 capturas de patentes del formato de 1995 y 67 imágenes sin ellas de 200x54 píxeles y 47 patentes del Mercosur y 114 imágenes sin ellas de 170x35 píxeles. Este no es un conjunto de datos grande, pero es suficiente para obtener resultados decentes para llevar a cabo el prototipo de este proyecto.

Todas estas imágenes se cargan en un programa proporcionado por uno de los autores del libro “Mastering OpenCV with Practical Computer Vision Projects”, el cual luego de procesar las muestras almacena los datos en una matriz  $N \times M$ , con  $N$  muestras y  $M$  características. Cada conjunto de datos se guarda como una fila en la matriz de entrenamiento. Una vez entrenado el sistema, se genera un archivo XML donde se encuentra almacenada la matriz con todos los datos procesados. Este archivo se incluye al proyecto de detección de patentes, para que al momento de capturar una nueva foto, la aplicación pueda recurrir a él y de esta forma lograr determinar cuáles de las posibles regiones dentro de la imagen corresponden a una patente y cuáles no, de esta manera se realiza el proceso de clasificación y se logra extraer solo aquellas regiones que contienen patentes.

En el diagrama 5 se observa el esquema que permite identificar al SVM dentro de la estructura del programa de detección.

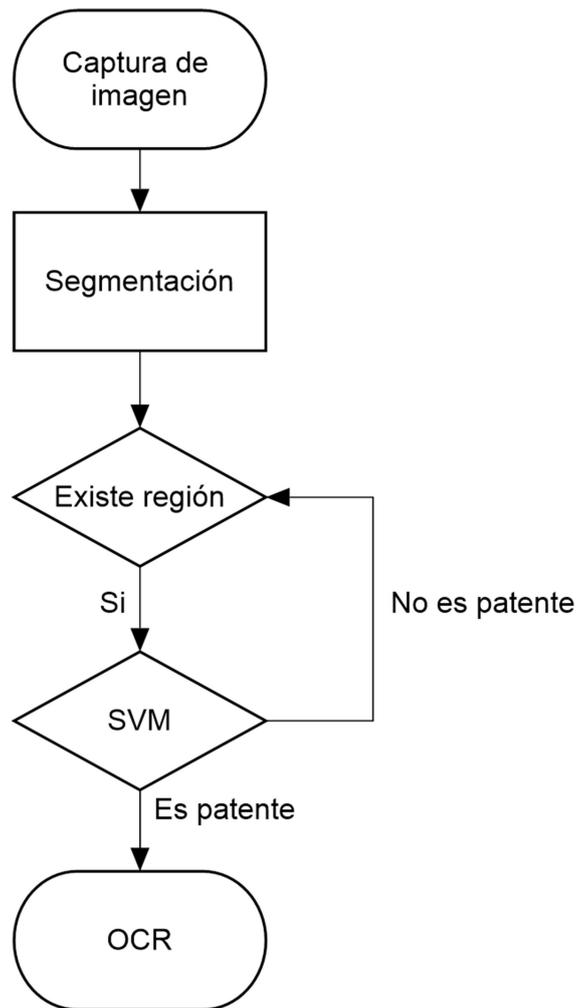


Diagrama 5: Flujo del SVM.

### Reconocimiento de caracteres [1]

El segundo paso en el reconocimiento de patentes es recuperar los caracteres de la misma con el OCR.

Por cada patente detectada, se procede a segmentar la placa para cada carácter, y utilizar un algoritmo de aprendizaje de una Red Neuronal Artificial (RNA) para reconocer el carácter.

En primer lugar, se debe preparar la imagen obtenida del SVM para que la aplicación pueda trabajar sobre esta para encontrar los caracteres, para ello es necesario convertir la imagen de RGB a escala de grises, aplicar un filtro de desenfoque gaussiano para eliminar ruido, aplicar un filtro de umbral para obtener una imagen binaria y usar esta imagen de umbral para detectar los contornos de los caracteres. Se puede ver este proceso en la Figura 14:

**AB 123 CD** Patente segmentada.

**AB 123 CD** Imagen de entrada en escala de grises y con filtro de desenfoque gaussiano.

**AB 123 CD** Luego de aplicar un filtro umbral para obtener una imagen binaria.

**AB 123 CD** Contorno de los caracteres encontrados.

**BVB 792** Patente segmentada.

**BVB 792** Imagen de entrada en escala de grises y con filtro de desenfoque gaussiano.

**BVB 792** Luego de aplicar un filtro umbral para obtener una imagen binaria.

**BVB 792** Contorno de los caracteres encontrados.

Figura 14: Imágenes filtradas de patentes para detectar sus caracteres.

En la imagen anterior se puede observar las transformaciones que sufre la imagen segmentada para lograr la identificación de los caracteres, siendo más evidente en la patente del formato de 1995, al convertir la imagen a escala de grises y aplicar el filtro gaussiano se logra reducir el ruido, en el caso de las patentes del formato 1995 se considera ruido a los puntos blancos que se encuentran sobre el fondo negro. Al aplicar el filtro umbral lo que se logra es ensanchar los caracteres y establecer un buen

contraste entre estos y el fondo, a partir de la imagen obtenida en este momento, se localizan las posibles regiones que contienen los caracteres, con la ubicación de estas se realiza el reconocimiento y lectura de los dígitos de la patente.

### Clasificación OCR [1]

En el paso de clasificación, se utiliza un algoritmo RNA de aprendizaje. Más específicamente, un Perceptrón de múltiples capas (MLP, por sus siglas en inglés), que es el algoritmo de RNA más utilizado.

Un perceptrón puede entenderse como la neurona artificial o unidad básica de inferencia en forma de discriminador lineal, a partir de lo cual se desarrolla un algoritmo capaz de generar un criterio para seleccionar un sub-grupo a partir de un grupo de componentes más grande.

Los MLP consisten en una red de neuronas con una capa de entrada, capa de salida y una o más capas ocultas. Cada capa tiene una o más neuronas conectadas con la capa anterior y la siguiente.

El siguiente ejemplo representa un perceptrón de 3 capas (es un clasificador binario que mapea una entrada vectorial de valor real a una única salida de valor binario) con tres entradas, dos salidas y la capa oculta incluyendo cinco neuronas:

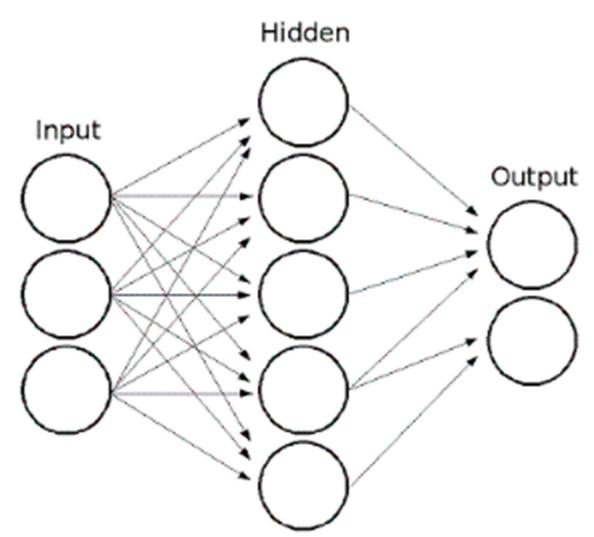


Figura 15: Ejemplo de perceptrón de 3 capas.

Todas las neuronas en un MLP son similares y cada una tiene varias entradas (las neuronas vinculadas anteriores) y varios enlaces de salida con el mismo valor (las

neuronas vinculadas siguientes). Cada neurona calcula el valor de salida como una suma de las entradas ponderadas más un término de polarización y se transforma mediante una función de activación seleccionada. En la Figura 16 se observa el funcionamiento de una neurona.

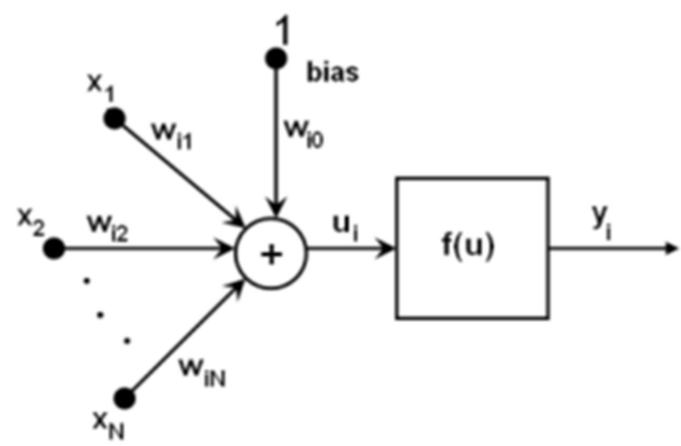


Figura 16: Funcionamiento general de una neurona.

Una red entrenada por RNA tiene un vector de entrada con características. Este pasa los valores a la capa oculta y calcula los resultados con los pesos y la función de activación. Luego, pasa sus salidas más abajo hasta que obtiene la capa de salida que tiene el número de clases de neuronas.

El peso de cada capa, sinapsis y neurona se calcula y aprende entrenando el algoritmo de RNA.

Para entrenar el clasificador, se crean dos matrices de datos como se hizo en el entrenamiento SVM, pero las etiquetas de entrenamiento son diferentes. En lugar de una matriz de  $N \times 1$  donde  $N$  representa las filas de datos de entrenamiento y 1 es la columna, se usa el identificador de número de etiqueta. Se tiene que crear una matriz  $N \times M$  donde  $N$  es el entrenamiento/muestras de datos y  $M$  es las clases (10 dígitos + 26 letras en este caso).

Para realizar el entrenamiento de esta sección, se tuvo que extraer todas las letras y números de las patentes del formato 1995, para la patente del Mercosur se utilizó la fuente Schriften a partir de la cual se basaron para la elaboración de las patentes, al no tener una gran diferencia entre los caracteres se obtuvieron excelentes resultados.

The image displays a large, bold, black font sample. The characters are arranged in three rows: the first row contains the uppercase letters A through M, the second row contains the uppercase letters N through Z, and the third row contains the digits 0 through 9. The font is a clean, sans-serif style with a consistent weight and spacing.

Figura 17: Tipografía Schriften.

En ambos caso se generó una imagen que contenía todo el abecedario y números del 0 al 9 en distintos tamaños, al cargar este archivo en el programa que genera el archivo OCR, programa proporcionado por uno de los autores del libro “Mastering OpenCV with Practical Computer Vision Projects”. Este programa produce todas las matrices necesarias y entrena el sistema, con la matriz de datos de entrenamiento, la matriz de clases y el número de neuronas en las capas ocultas.

Los datos de entrenamiento se cargan desde un archivo XML tal como se hizo para el SVM. En el Diagrama 6 se observa el esquema que permite identificar al OCR dentro de la estructura del programa de reconocimiento.

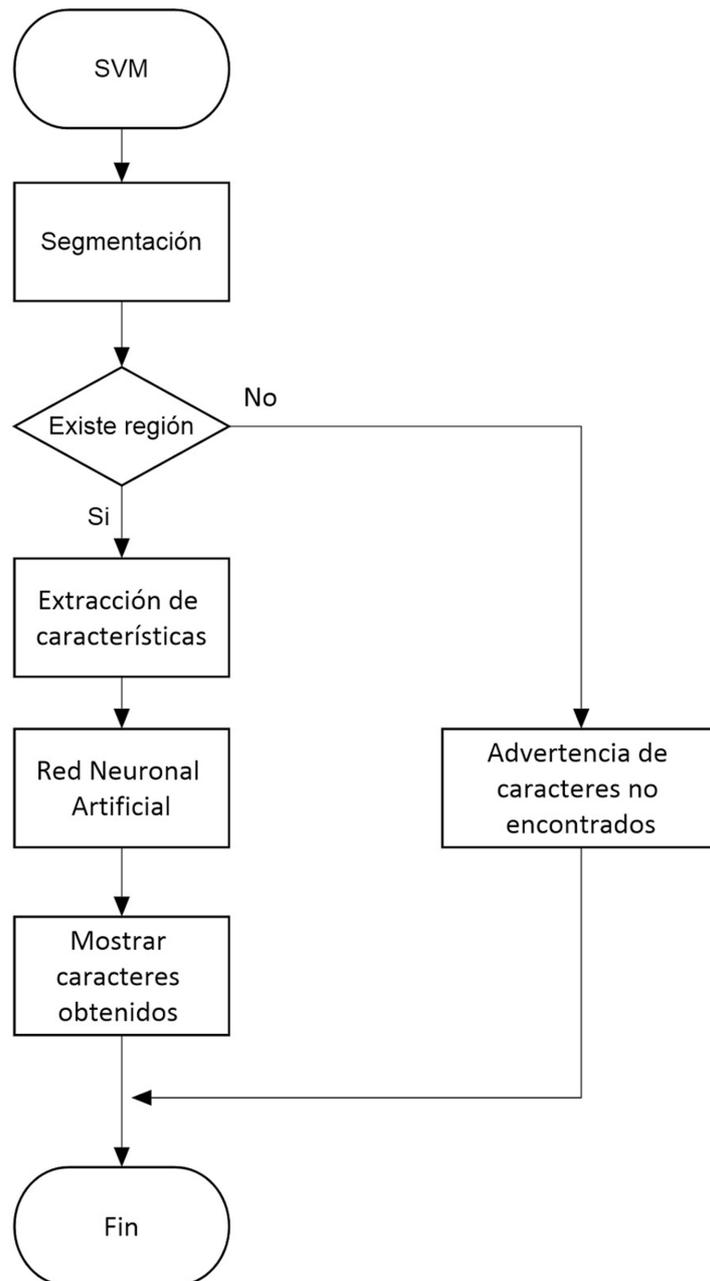


Diagrama 6: Flujo del OCR.

#### 2.2.4 Pruebas realizadas

A continuación se detallan algunos de los resultados obtenidos una vez finalizado el entrenamiento tanto del SVM como del OCR y luego de establecer los valores de los filtros de umbral y parámetros de las fotos para su procesamiento.

Los resultados obtenidos corresponden a distintas patentes de autos estacionados en la playa de estacionamiento de la UTN FRP la cual se encuentra al aire libre, las fotografías no fueron tomadas con la cámara que finalmente se utilizó para el prototipo

aunque los parámetros utilizados para el procesamiento de imágenes se mantuvieron debido a que los resultados obtenidos fueron óptimos e independientes de la calidad con la que se captura la fotografía.

Se debe aclarar que se hicieron algunas correcciones por software al momento de la lectura de caracteres, debido a la similitud que presentan algunas letras y números, como por ejemplo en las patentes con formato de 1995 el caso de la “B” y el “8”, antes de determinar de qué carácter se trata se evalúa si el mismo corresponde a un número o una letra, dependiendo su ubicación dentro de la patente, además, en el caso de que no se puedan identificar la totalidad de los caracteres, es decir, 7 para el formato Mercosur (4 letras y 3 números) y 6 para el formato de 1995 (3 letras y 3 números) se descarta lo reconocido y se informa al operario que no se pudo leer la patente y la misma debe ser ingresada de forma manual.

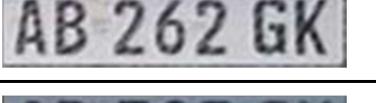
Patente segmentada	Caracteres reconocidos	Caracteres extraídos
		BVB792 - Correcto
		PDN454 - Correcto
		No se detectó caracteres
		GDF304 - Correcto
		LAC482 - Falló
		No se detectó caracteres
		AB262GK - Correcto
		AB385QN - Falló

Tabla 1: Prueba de procesamiento de imágenes

Los resultados mostrados en la tabla anterior solo son de algunas de las pruebas realizadas.

Con patentes del Mercosur se probó el sistema con un total de 52 patentes, de las cuales en 39 casos se reconoció y registró correctamente las placas y sus caracteres, en 9 casos no se logró detectar y en 4 oportunidades si bien se extrajo la patente de manera correcta presentaron fallas en el reconocimiento de los caracteres.

Para las patentes con el formato de 1995 se probó el sistema con 95 patentes, de las cuales en 45 casos se reconoció y registró correctamente las placas y sus caracteres, en 19 casos no se logró detectar y en 30 oportunidades si bien se extrajo la patente de manera correcta presentaron fallas en el reconocimiento de los caracteres. Para este último caso, en un 80% de las veces el error se produjo en un solo carácter.

### **2.2.5 Cámara**

En la captura de fotos de patentes para realizar el entrenamiento del clasificador SVM se utilizó la cámara de un teléfono celular de 8[Mp], para este proceso lo más importante era la distancia a la que se tomaba la foto y el tamaño de la patente dentro de la imagen, por lo que las capturas realizadas con esta cámara cumplieron con éxito el objetivo.

Para la aplicación, donde se requiere detectar la patente al momento de ingreso y egreso de un vehículo, la exigencia en la calidad de imagen toma importancia. En una primera instancia se probó utilizar una cámara web común y una de alta calidad, con resultados desfavorables, posteriormente, aprovechando las prestaciones del teléfono celular y mediante una aplicación para el mismo que permite simular el uso de una cámara IP, se realizaron pruebas con este medio, si bien los resultados fueron positivos con respecto a los obtenidos con las webcam, seguía siendo insuficiente para las exigencias de la aplicación. Finalmente se consiguió realizar pruebas con cámaras IP nativas, una de interiores de 1[Mp] y una de exteriores de 2[Mp]. En el primer caso los resultados fueron óptimos, logrando para una determinada configuración de la cámara, detectar patentes que se encontraban a 3[m] de distancia, aunque en mucho de los casos fallaba al momento de reconocer los caracteres de la misma. La cámara IP de exteriores además de tener más megapíxeles de resolución, permite trabajar en modo infrarrojo permanentemente, lo cual es una ventaja para realizar el

reconocimiento de patentes. Con esta cámara se logró detectar patentes ubicadas a 3[m] con un buen reconocimiento de los caracteres de las mismas, tanto en las patentes del Mercosur como en las del formato de 1995.

Una característica a tener en cuenta al momento de instalar la cámara es la de configurarla para el entorno donde va a trabajar, debido a que las condiciones de luz y ambiente no son las mismas en un lugar cerrado o semicerrado que al aire libre, en este último la luminosidad va a depender de las condiciones climatológicas, no es lo mismo un día de sol que uno nublado o de lluvia. En lugares cerrados, las condiciones de luz van a depender de la iluminación del lugar y permanecerá constante a lo largo del día. Para este proyecto la cámara se configuró para un lugar cerrado e iluminado por tubos fluorescentes, esto último se tiene en cuenta para configurar el balance de blanco. El sistema se puede implementar en otras condiciones, como ser en un lugar al aire libre, siempre y cuando se configure previamente la cámara. En la Figura 18 se observan todos los parámetros que se pueden configurar de los ajustes internos de la cámara.

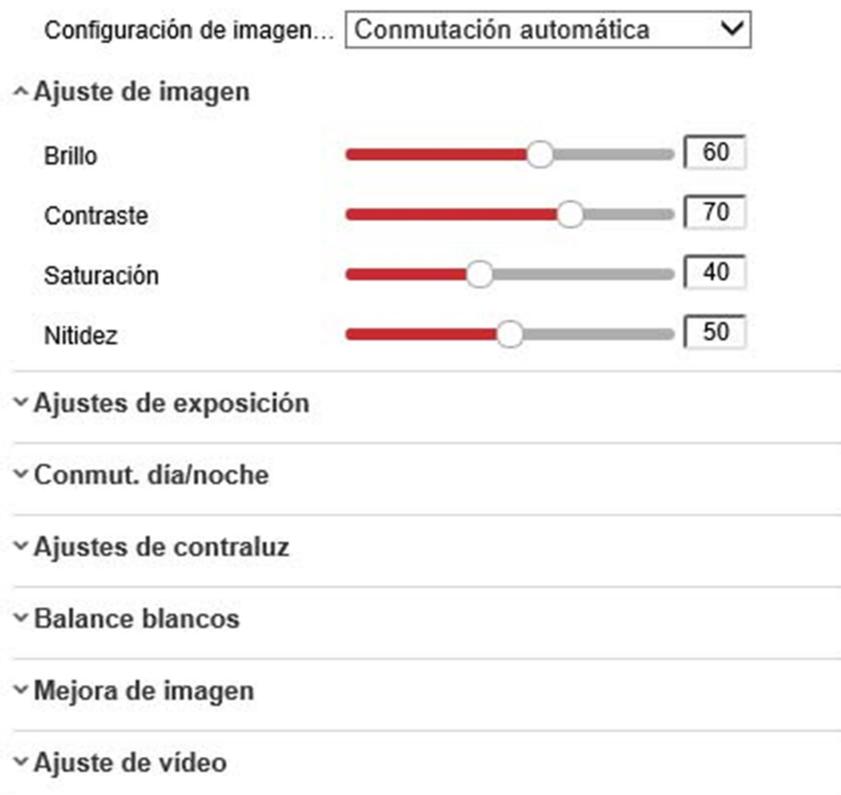


Figura 18: Configuración de imagen de la cámara IP.

Los ajustes más importantes a tener en cuenta al momento de configurar la cámara son, el brillo, contraste, saturación, ajustes de contraluz en caso de que se instale en el exterior y pueda ser afectada por el sol de frente, el balance de blanco para lograr que los colores no se vean alterados y en caso de que sea posible configurar la cámara en modo noche para que trabaje en infrarrojo permanentemente lo que es un beneficio para los fines del proyecto.

#### Diferencias entre cámaras IP nativas y analógicas [2]

Una cámara analógica es una cámara tradicional utilizada en los sistemas CCTV. Envía video sobre cable a VCRs o DVRs. Las cámaras IP son todas las cámaras digitales que pueden enviar imágenes, por cable, directamente a la red. Muchos de los sistemas de cámara de seguridad de hoy en día son sistemas híbridos que incorporan tanto componentes analógicos como digitales.

Las cámaras IP poseen una calidad de video superior a las cámaras analógicas y ofrecen más posibilidades de visualización, tales como ampliar o estrechar el campo de visión y mejor capacidad de zoom. Debido a que la transmisión de las señales no requieren de compresión, proporcionan más detalles y hacen que sean mucho mejor para el reconocimiento facial o la detección de matrículas.

Las cámaras analógicas tienen menor calidad, pero trabajan mejor en condiciones de poca luz, tienen más limitadas las posibilidades de visualización y no ofrecen la calidad de zoom de las cámaras IP. Si se aplica un zoom se obtendrá una imagen más granulada y degradada.

Las cámaras analógicas se limitan a resoluciones estándar NTSC/PAL de 720x480 píxeles (NTSC) / 575 (PAL) o 0,4[Mp] (4CIF). Las cámaras IP ofrecen resoluciones que van desde 1,3[Mp] hasta 5[Mp] (2560x1920) comprimidas y codificadas. Esto permite cubrir un área mucho más amplia de observación y obtener imágenes más detalladas al ampliar la misma.

Las cámaras analógicas tradicionales operan sobre cable coaxial, Para poder ver una cámara analógica a través de la red o del internet esta debe ser conectada a un DVR. Las cámaras IP pueden trabajar sobre par trenzado y conexiones inalámbricas.

En algunos casos las cámaras IP pueden ser alimentadas por el cable Ethernet de par trenzado (alimentación PoE), eliminando preocupaciones sobre realizar tendidos de cables eléctricos. Esto no sucede con las cámaras analógicas.

Las cámaras analógicas pueden enviar video sobre cable de par trenzado hasta 1,5[km] de distancia y hasta 300[m] sobre cable coaxial, pero las transmisiones pierden calidad en distancias largas o cuando la señal es convertida en formatos diferentes.

Las cámaras IP pueden enviar datos a 100[m] por cable Ethernet de par trenzado y a distancias ilimitadas mediante redes IP.

### 2.3 Aplicación de PC

La interfaz de usuario fue realizada en el entorno de desarrollo integrado Microsoft Visual Studio 2017 para sistemas operativos Windows.

La aplicación registra el ingreso y egreso de un vehículo, detectando la patente del mismo para así poder calcular el monto a cobrar y el tiempo que utilizó el servicio, también permite observar de forma rápida el estado de ocupación de la playa, además lleva un registro de los ingresos de autos, motos e ingresos económicos para poder llevar un control de uso. Esta aplicación le permite al operario poder establecer las tarifas para autos y motos, e indicar que patente realizó un pago mensual.

Dado que la aplicación puede cerrarse, es necesario que los datos almacenados en la configuración no se pierdan, para ello se guardan en archivos de textos que se actualizan durante el funcionamiento del programa cuando se produce algún cambio o evento. Cuando se abre la interfaz, se cargan automáticamente los datos guardados en distintos archivos y de no encontrarse, se cargan por defecto con valores predeterminados.

Los archivos en cuestión son:

- *Configuracion.txt*: Contiene los valores de todas las tarifas y de los horarios de apertura y cierre de la playa.
- *DatosEstadisticas.txt*: Contiene los datos estadísticos acumulados.
- *PatentesReservadas.txt*: Contiene las patentes que han pedido una reserva mensual en la playa.
- *RegistroAutos.txt*: Contiene las patentes de los autos que están actualmente en la playa junto con la hora de ingreso de cada una.

- *RegistroMotos.txt*: Contiene las patentes de las motos que están actualmente en la playa junto con la hora de ingreso de cada una.
- *Reserva.txt*: Contiene el estado (libre, ocupado o reservado) de los 6 lugares que dispone el prototipo, de manera que al iniciarse la aplicación se puedan establecer los estado de cada UR.

La interfaz consta principalmente de tres pestañas: Inicio, Configuración y Estadísticas.

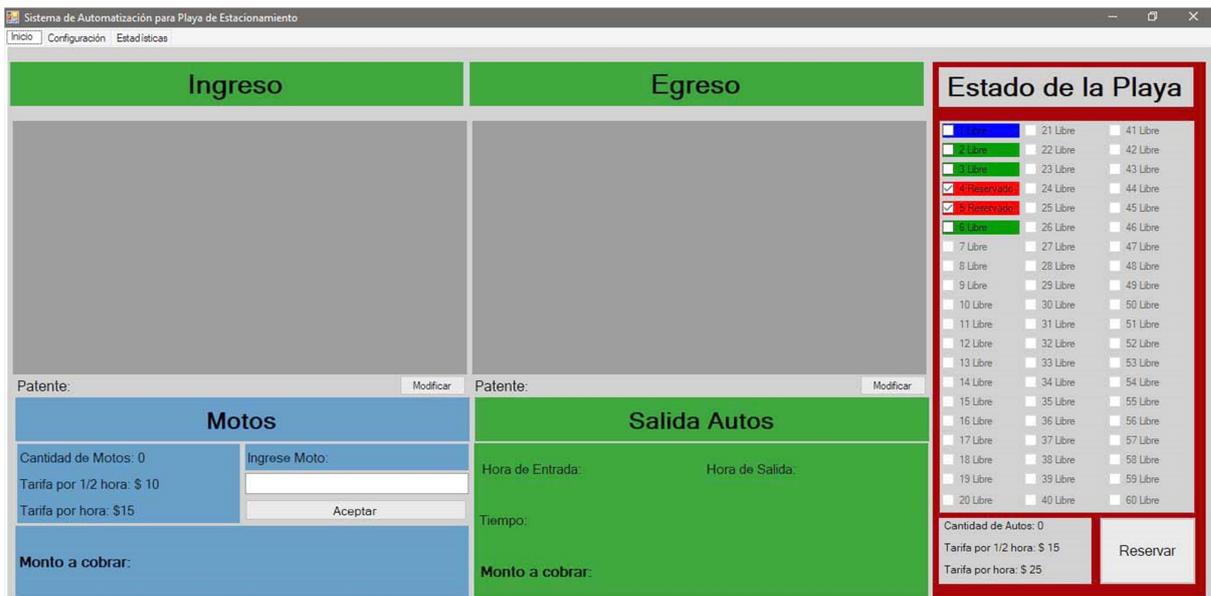


Figura 19: Pestaña de Inicio.

Se pueden destacar tres partes principales de la pestaña de Inicio:

- **Motos**: Ubicada en la parte inferior izquierda y en azul. En el cuadro de texto blanco se ingresa la patente o alguna identificación de la moto. La primera vez que se ingresa la registra en la base de datos, si posteriormente se vuelve a ingresar la misma patente se procede al cobro indicando el monto a cobrar. También se indican la cantidad de motos en el estacionamiento, la tarifa por media hora y por hora. Cuando ingresa una moto se guarda en el archivo *RegistroMotos.txt* la patente o identificación escrita por el operario y la hora de ingreso, cuando egresa la moto se lee de este archivo para calcular el tiempo que estuvo en el estacionamiento y mostrar el monto a cobrar según la tarifa, posteriormente se borran los datos correspondientes a esa moto en el archivo.
- **Autos**: En verde. Tiene dos cuadros de imágenes en los cuales se visualizan las fotos tomadas por la cámara IP en el ingreso y egreso de los vehículos. Al

momento de detectar un vehículo ingresando en la playa, la cámara realiza una captura para proceder con el procesamiento digital de la imagen y visualizarla en pantalla (cuadro de imagen izquierdo). Una vez tomada la foto del vehículo se procede a la detección de la patente, primeramente se busca una con el formato de 1995, de no encontrarla se busca una del Mercosur y en caso de otro resultado, aparece en la pantalla una subventana que le indica al operario que no se detectó ninguna patente y le da la opción de ingresarla manualmente o cancelar el evento en caso de que se haya activado por otro motivo el sensor encargado de la detección de la presencia de un vehículo. Si se logra detectar una patente, debajo de la imagen aparece el código de la misma y se ingresa automáticamente a la base de datos. En caso de que el código sea erróneo, el usuario cuenta con el botón “Modificar” (debajo de la esquina inferior derecha de la imagen) para corregirlo. Una vez ingresada la patente, se guarda en el archivo *RegistroAutos.txt* su código y hora de ingreso. En el momento de egreso de un vehículo, el procedimiento es el mismo, solo que en este caso, el código de la patente se busca en el archivo junto con su hora de ingreso para proceder con la tarificación y posteriormente se eliminan estos datos del mismo. Si la patente se encuentra reservada, es decir, el cliente realizó un pago mensual por el alquiler de la playa en vez de figurar un importe en el monto a cobrar, aparecerá una leyenda “Alquiler Mensual”. En recuadro “Salida Autos” aparece la fecha y hora de entrada, y la de salida, el tiempo que estuvo adentro el vehículo y el monto a cobrar.

- Estado de la Playa: Ubicada a la derecha. En esta parte se observa el estado de ocupación de los lugares de estacionamiento. El-prototipo solo cuenta con los primeros seis lugares pero en la interfaz se observa la totalidad de la playa, recordando que para este proyecto se pensó en una playa con 60 lugares para estacionar (los lugares del 7 al 60 están deshabilitados). Cada lugar cuenta con una casilla, mediante la cual se puede reservar o no el lugar según si esta se encuentra tildada o no. El texto y el color de la casilla cambian según el estado del lugar. El primer lugar por ejemplo, se encuentra con un fondo azul indicando que su uso es para personas con capacidades diferentes, y aparece en “Libre”, al momento de que un vehículo se estaciona, el color de fondo cambia a rojo y su

texto a “Ocupado”. Para los demás lugares, cuando están vacíos el texto es “Libre” y el fondo verde, cuando se estaciona un auto cambian a “Ocupado” y rojo, y en caso de reserva a “Reservado” y rojo respectivamente. Debajo de los lugares se encuentran etiquetas que indican la cantidad de autos estacionados, la tarifa por media hora y por hora, y el botón de “Reservar”, el cual cumple la función de comunicarle a las URs si deben reservar su lugar. Al clickearlo guarda en el archivo *Reserva.txt* el estado de cada uno de los 6 lugares del prototipo-

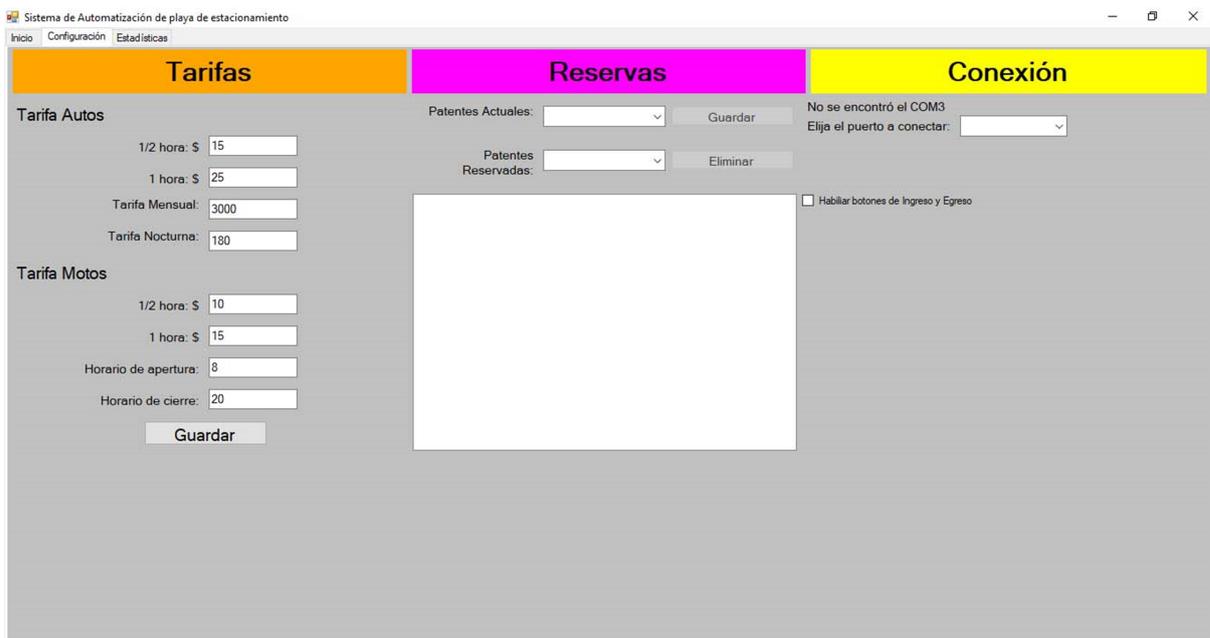


Figura 20: Pestaña de Configuración.

Se pueden destacar tres partes principales de la pestaña de Configuración:

- **Tarifas:** En esta parte se encuentran las tarifas de los autos (por media hora, hora, mensual y nocturna), las de las motos (por media hora y hora) y los horarios de apertura y cierre de la playa. Se puede observar que junto a cada etiqueta se encuentra un cuadro de texto en el que se puede observar el valor correspondiente. En este lugar el operario puede modificar las tarifas y los horarios. Para esto último se utiliza el botón “Guardar”, que al clickearlo guarda todos los cambios realizados y actualiza el archivo *Configuración.txt*.
- **Reservas:** En esta parte se pueden observar las patentes que están actualmente en la playa y las reservadas. Primeramente, en la parte superior se encuentran las patentes que se encuentran actualmente dentro de la playa, para observarlas se debe clikear sobre el desplegable. Este da la posibilidad de

seleccionar una de las patentes para reservar mediante el botón “Guardar” que se encuentra a la derecha, al presionar el botón se guarda la patente en el archivo *PatentesReservadas.txt* y su código aparece en la lista de texto (cuadro blanco). En el desplegable de abajo, se muestran las patentes reservadas, pudiendo eliminar la reserva de alguna de ellas al seleccionarla y clicar el botón “Eliminar”. Al hacer esto último, se borra la patente del archivo y su código de la lista de texto.

- **Conexión:** En esta parte se establece la conexión con el puerto que conecta a la UC con la PC. El programa intenta hacerlo automáticamente y de no lograrlo lo informa en una etiqueta. Al hacer click en el botón desplegable aparecen todos los puertos disponibles y seleccionando uno de ellos se establece la conexión con el mismo. Debajo se encuentra una casilla que es utilizada para habilitar botones ocultos en la pestaña de Inicio que sirven para emular los eventos de ingreso y egreso de un vehículo sin necesidad de que estén conectadas la UC y las URs.

	Hoy	Ayer	Últimos 7 días	Últimos 30 días	Últimos 90 días
Cantidad de Autos	0	77	77	77	77
Total Autos [\$]	0	10970	10970	10970	10970

	Hoy	Ayer	Últimos 7 días	Últimos 30 días	Últimos 90 días
Cantidad de Motos	0	0	0	0	0
Total Motos [\$]	0	0	0	0	0

	Hoy	Ayer	Últimos 7 días	Últimos 30 días	Últimos 90 días
TOTAL [\$]	0	10970	10970	10970	10970

Figura 21: Pestaña de Estadísticas.

En la pestaña de Estadísticas podemos observar los datos como la cantidades de autos y motos, los ingresos económicos de cada caso y el ingreso total; en el día actual (Hoy), el anterior (Ayer) y los acumulados en períodos de una semana (Últimos 7 días), un

mes (Últimos 30 días) y un trimestre (Últimos 90 días). Todos estos datos están guardados en el archivo *DatosEstadisticas.txt* y se actualizan constantemente.

### 2.3.1 Pruebas de la Aplicación de PC

Las pruebas de la interfaz se realizaron para todas las posibilidades de funcionamiento y con los dos tipos de patentes desarrollados a lo largo del proyecto. Se comentan las siguientes imágenes con los casos correspondientes:

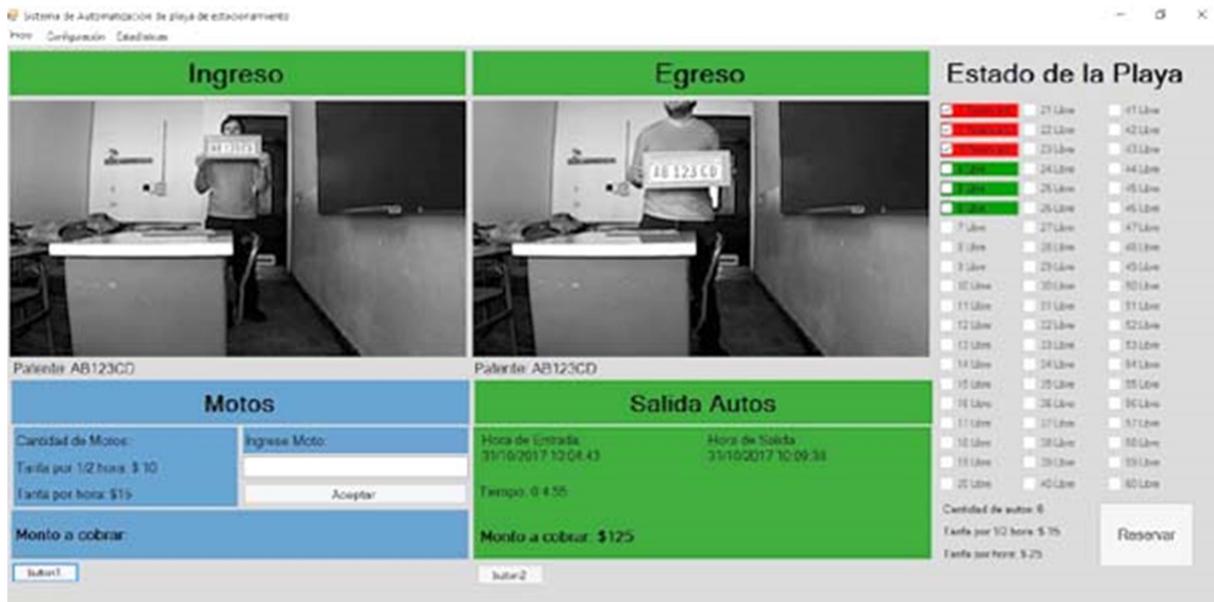


Figura 22: Prueba N° 1.

Prueba N° 1: Se utilizó una patente del Mercosur, la imagen de ingreso se tomó a 3[m] de la cámara y la de egreso a 2[m]. En ambos casos se detectó correctamente la patente y el programa calculó el tiempo y el monto a cobrar correctamente.

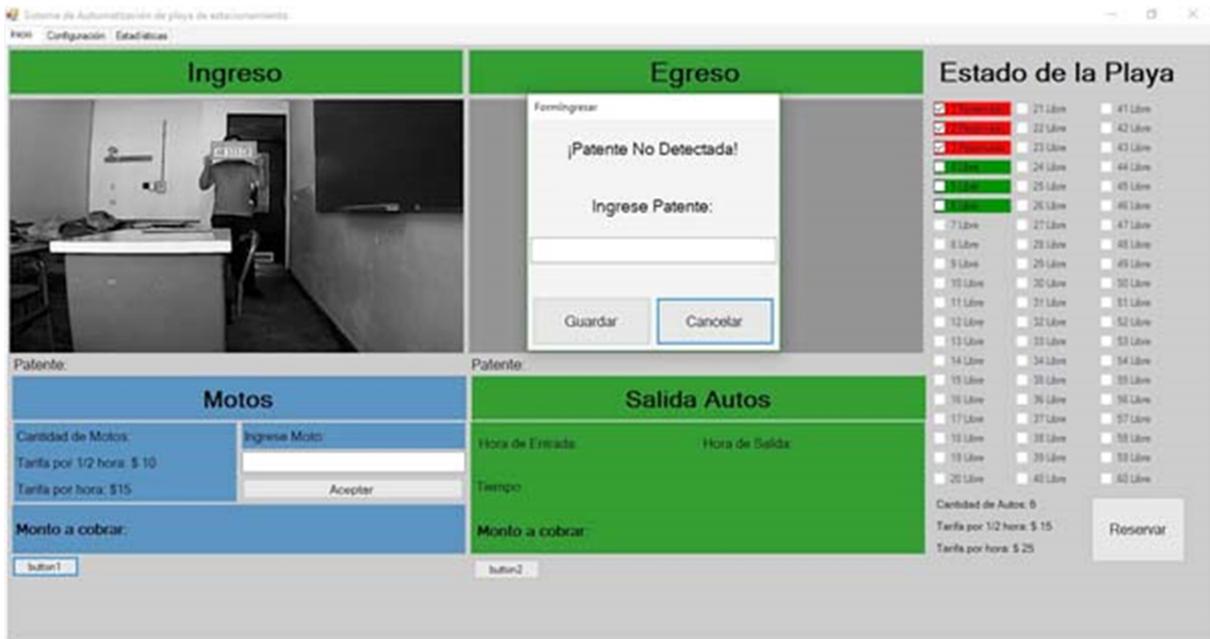


Figura 23: Prueba N° 2.

Prueba N° 2: Se utilizó una patente del Mercosur, la imagen de ingreso se tomó a 4[m] de la cámara. Se observa la subventana, la cual nos indica que no se pudo detectar la patente y nos da la opción de ingresarla en el cuadro de texto o de cancelar el evento si fuera fallido.

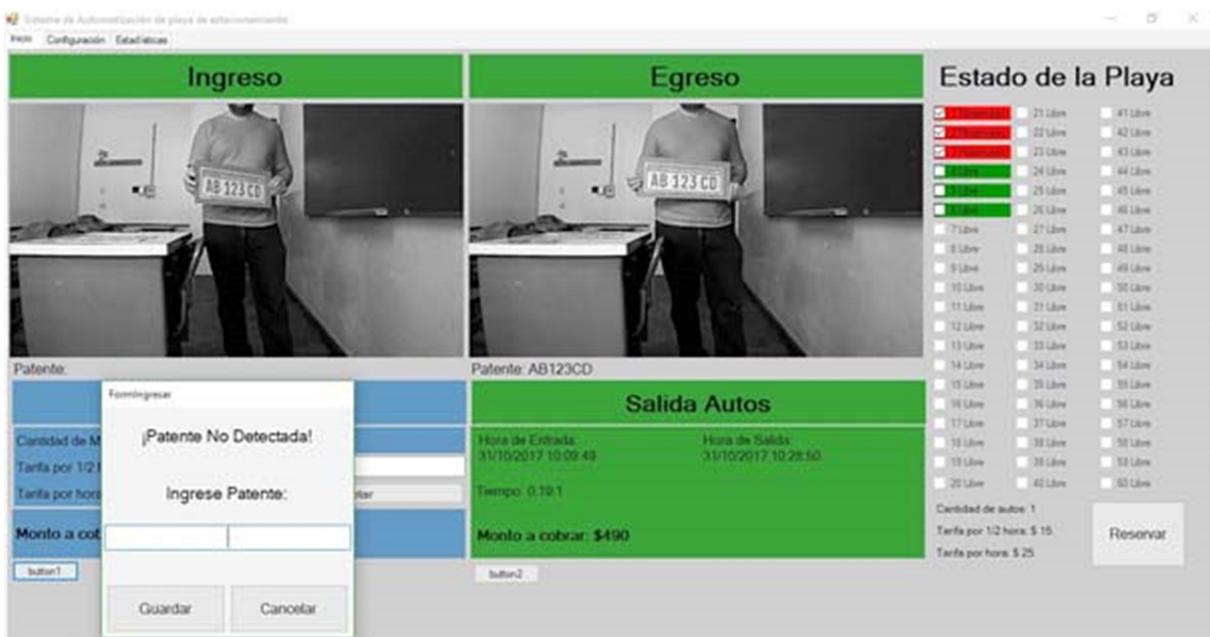


Figura 24: Prueba N° 3.

Prueba N° 3: Se utilizó una patente del Mercosur, la imagen de ingreso se tomó con una perspectiva de profundidad de 45° a 1,5[m] de la cámara y la de egreso a la misma

distancia pero a una perspectiva menor. En el primer caso no se detectó la patente y en el segundo si.



Figura 25: Prueba N° 4.

Prueba N° 4: Se utilizó una patente del Mercosur, la imagen de egreso se tomó a 1,5[m] de la cámara y con una inclinación de 45° (diagonal). El sistema no pudo detectar la patente.

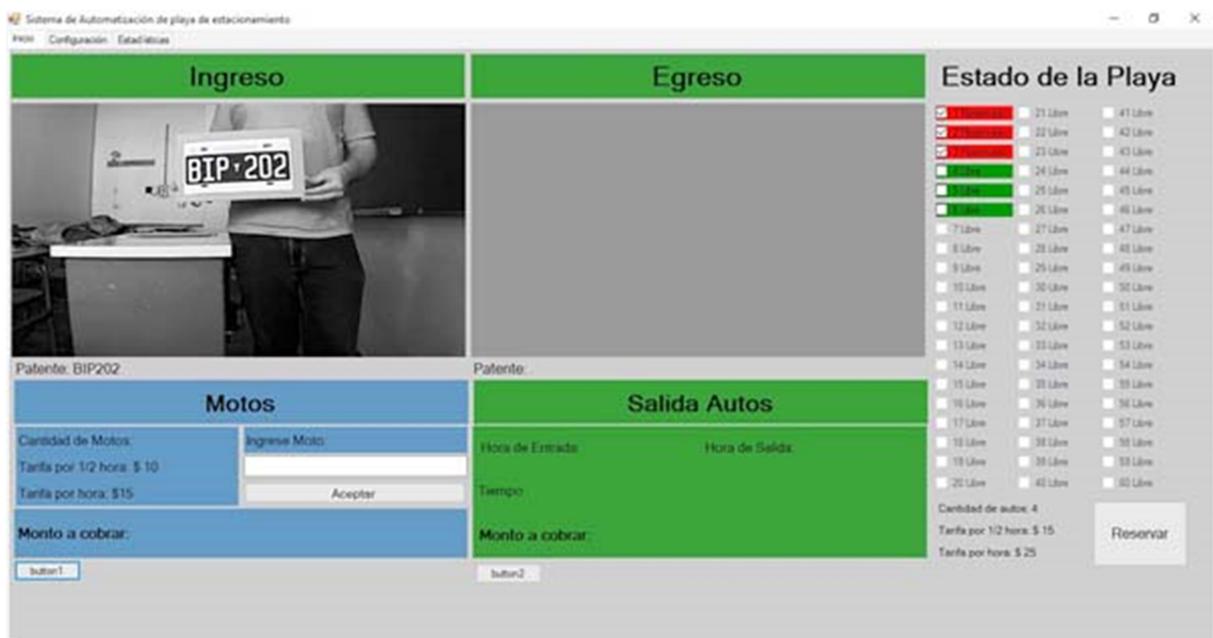


Figura 26: Prueba N° 5.

Prueba N° 5: Se utilizó una patente de formato de 1995, la imagen de ingreso se tomó a 1[m] de la cámara y se detectó correctamente la patente.

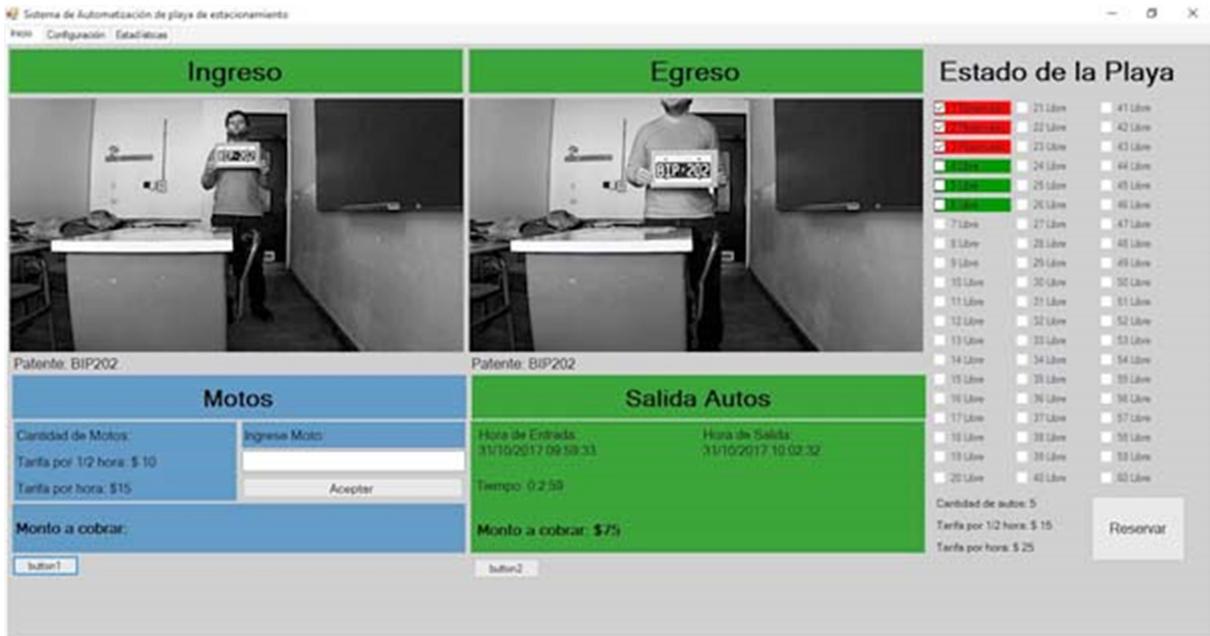


Figura 27: Prueba N° 6.

Prueba N° 6: Se utilizó una patente de formato de 1995, la imagen de ingreso se tomó a 3[m] de la cámara y la de egreso a 2[m]. En ambos casos se detectó correctamente la patente y el programa calculó el tiempo y el monto a cobrar correctamente.

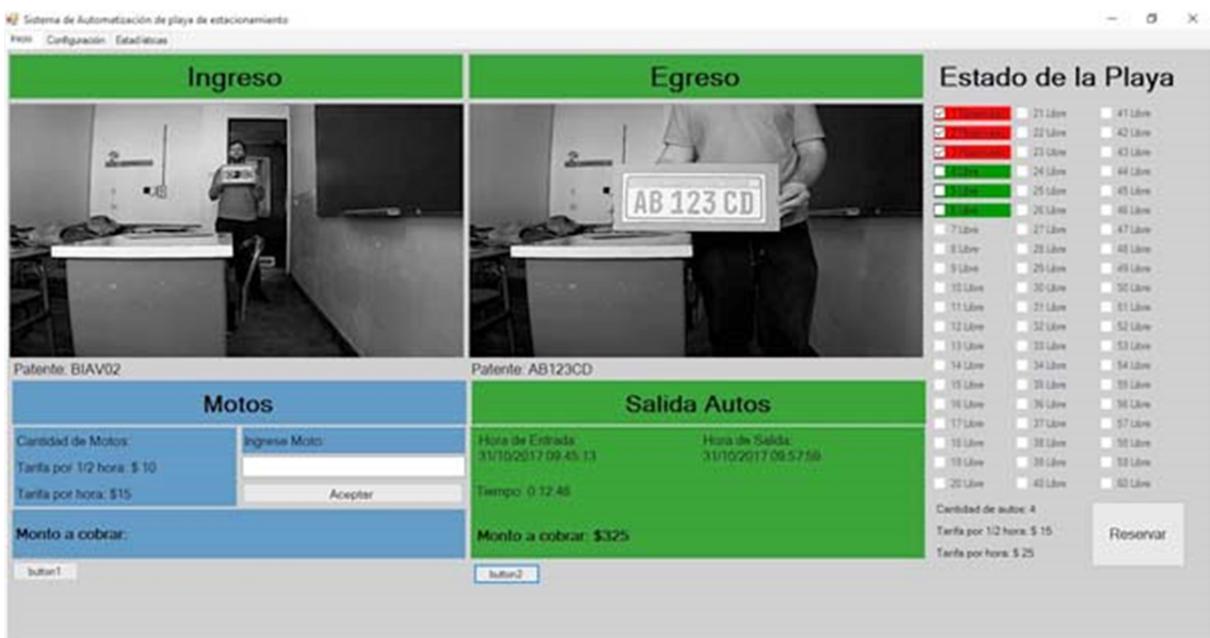


Figura 28: Prueba N° 7.

Prueba N° 7: Se utilizó una patente de formato de 1995 (BIP202) para la imagen de ingreso a 4[m] de la cámara y una del Mercosur para la de egreso a 1[m]. En el primer caso se detectó la patente pero el código mostrado fue erróneo (BIAV02). En el segundo caso se detectó correctamente la patente.

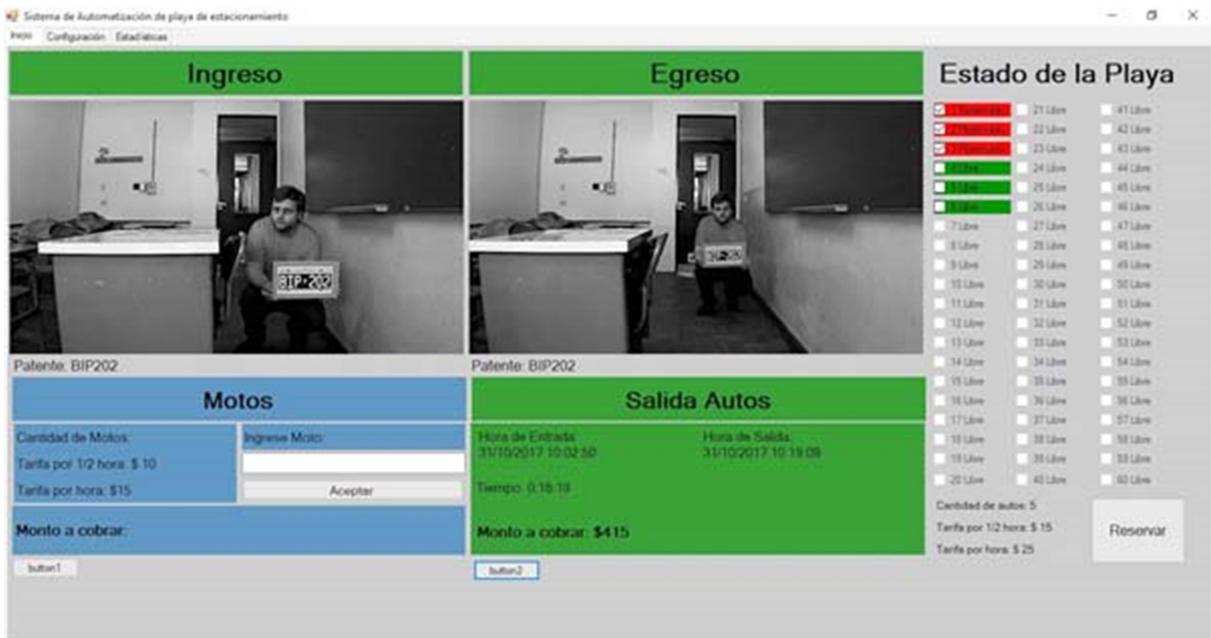


Figura 29: Prueba N° 8.

Prueba N° 8: Se utilizó una patente de formato de 1995, la imagen de ingreso se tomó a 2[m] de la cámara pero a una altura menor y la de egreso a 3[m]. En ambos casos se detectó correctamente la patente y el programa calculó el tiempo y el monto a cobrar correctamente.

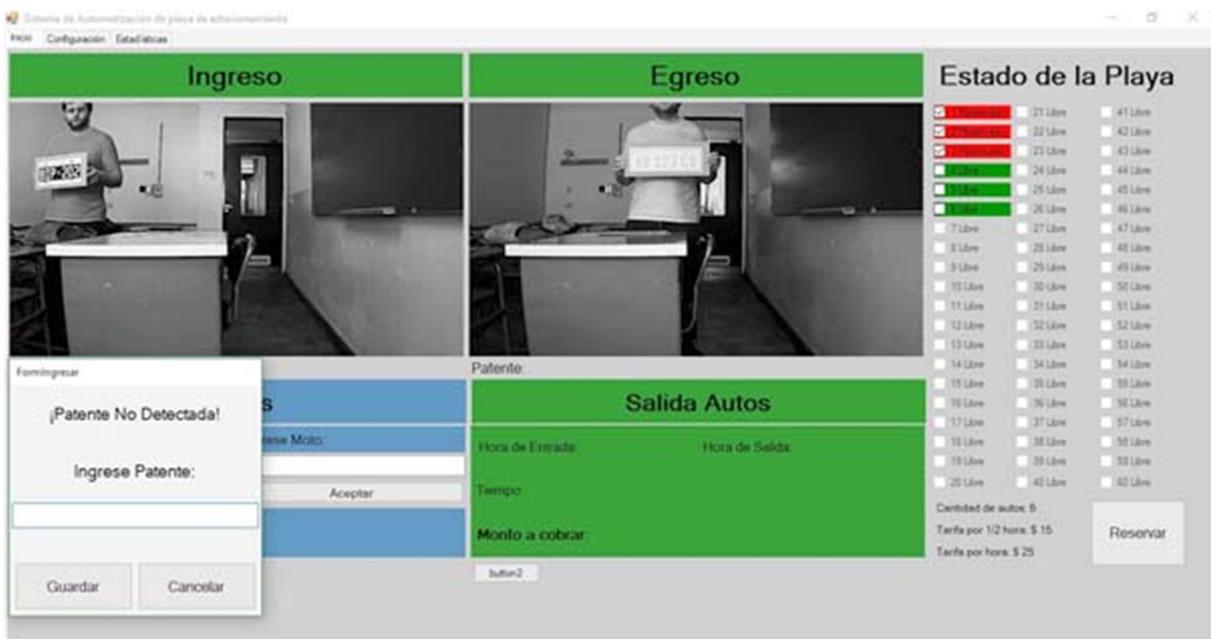


Figura 30: Prueba N° 9.

Prueba N° 9: Se utilizó una patente de formato de 1995 para la imagen de ingreso a un radio de 2[m] de la cámara pero no en línea recta a su lente y no se detectó la patente.

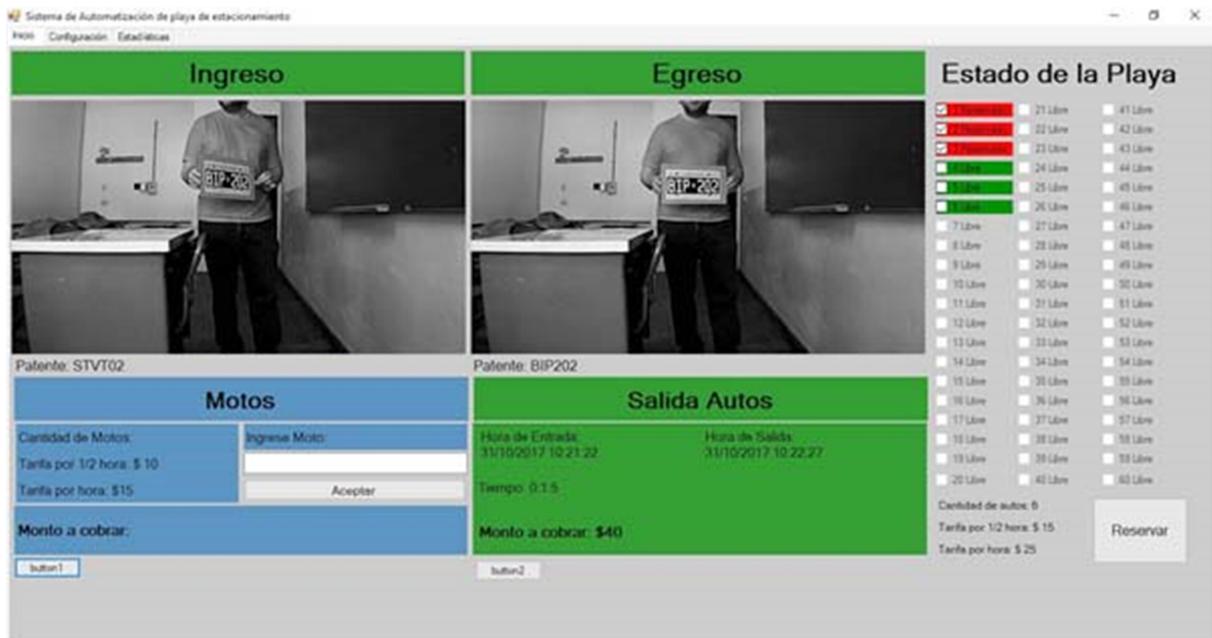


Figura 31: Prueba N° 10.

**Prueba N° 10:** Se utilizó una patente de formato de 1995 (BIP202), la imagen de ingreso se tomó con una perspectiva de profundidad de 45° a 1,5[m] de la cámara y la de egreso a la misma distancia pero a una perspectiva menor. En el primer caso se detectó la patente pero su código fue erróneo (STVT02) y en el segundo se detectó correctamente.

### 2.3.2 Problemas y Soluciones de la Aplicación de PC

En las pruebas pudo notarse que si se detectaba una patente en la entrada pero su código era incorrecto, este quedaba registrado pudiendo producirse que a la salida se detectara correctamente o incluso nuevamente equivocado pero con distinto código. Debido a esta incongruencia nunca se producía el cobro porque en el archivo *RegistroAutos.txt* nunca se encontraría el código detectado a la salida. Para solucionar esto se procedió a la inclusión de los botones “Modificar” a la entrada y la salida. Estos le dan la posibilidad al operario de corregir cualquier código incorrecto cuando se detecta una patente.

Para el reconocimiento de patentes, las condiciones de luz afectan mucho al momento de realizar la captura de la fotografía, por lo que se debe configurar la cámara para el entorno donde se instalará y los parámetros a modificar deben hacerse de manera empírica hasta obtener resultados óptimos. No solo las condiciones de luz afectan a la detección de patentes, otro inconveniente se presenta cuando la patente ingresada no

se encuentra paralela a la cámara, sino que tiene una cierta inclinación o perspectiva, en este caso es posible detectar la patente y sus caracteres cuando dicha distorsión no es muy pronunciada, en el prototipo implementado, no se realizó ningún algoritmo para corregir esto y se ingresa la patente de frente, pero en caso de implementar, se debe definir la posición donde se encontrará la cámara, donde ingresarán los vehículos y aplicar una transformación a la imagen para contrarrestar la perspectiva de la patente. Los ajustes a la imagen deben hacerse de manera empírica, para esto la cámara, el lugar de ingreso y egreso deben permanecer fijos.

#### **2.4 Diseño Completo (Prototipo)**

Para el montaje del prototipo se dispuso que cada UR esté dentro de una caja estanco al igual que la UC y que estén comunicadas entre si por medio de dos tubos de PVC rígidos. Los cables de alimentación alterna de 24[V] y 220[V] van por un mismo tubo y el par trenzado de comunicación por el otro.

Las cajas estanco se modificaron para poder incluirles los portafocos, el sensor y el fusible. La UC tiene el transformador que alimenta todas las unidades, dos pulsadores para simular cuando se detecta un vehículo a la entrada y a la salida y un cable USB para conectarse a la PC.

Para la realización del cartel de entrada se utilizaron LEDs de alto brillo tipo flat dado que presentan un haz luminoso de gran expansión y evita que se concentre la luz en un solo punto, mejorando así la visualización.

En la Figura 32 se observa la disposición de las URs donde cada unidad estaría ubicada sobre el lugar a sensar.



Figura 32: Unidades Remotas conectadas.

Se implementaron 6 unidades para el prototipo, separadas 60[cm] entre si, cada caja cuenta con un sensor para la detección de obstáculos y dos lámparas para indicar el estado de cada lugar. Para poder realizar pruebas y presentaciones las URs cambian de estado al detectar un objeto a menos de 1[m] de distancia.

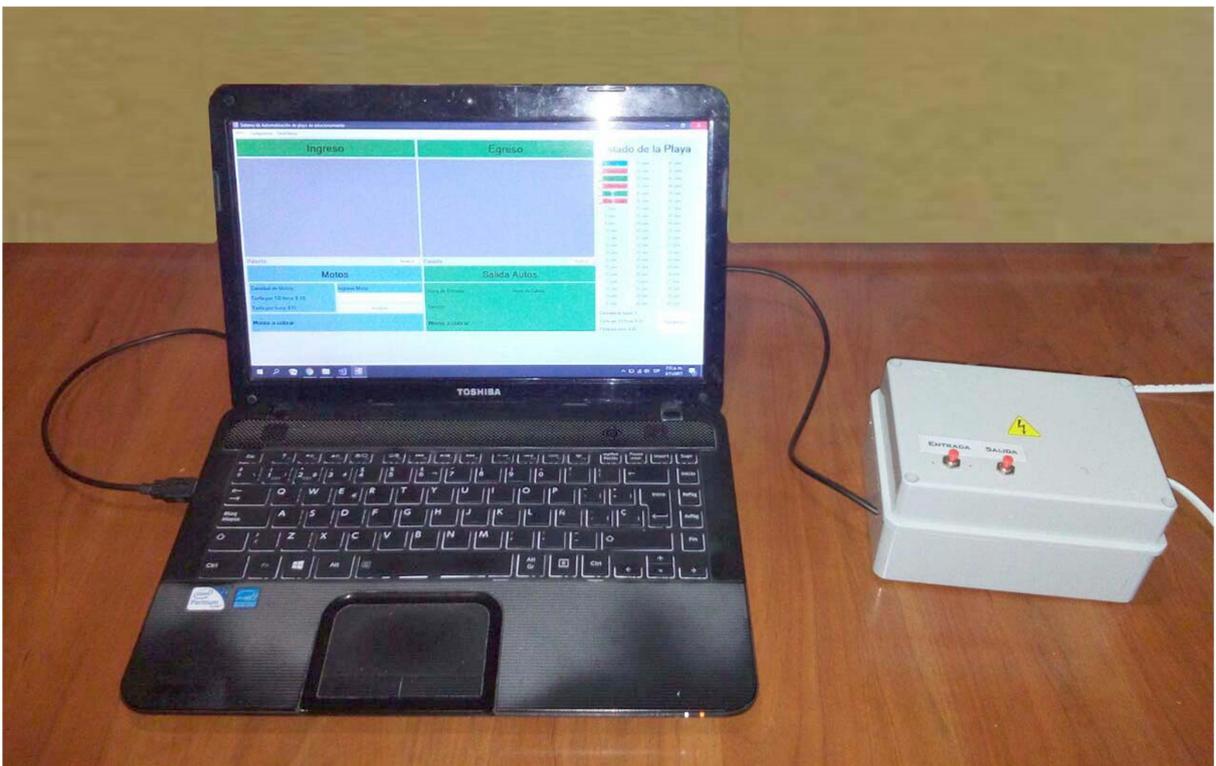


Figura 33: Aplicación de PC y Unidad Central.

La UC se conecta a la PC mediante un cable USB, la central realizará sus actividades únicamente cuando se encuentre conectada a la aplicación de PC.

En la pantalla principal de la aplicación se pueden observar todos los datos relacionados al uso de la playa, se pueden reservar lugares y observar los estados en los que se encuentran los mismos, los cuales se actualizan de manera inmediata, aunque figuran los 60 posibles lugares de la playa, solo se encuentran activos los primeros 6 que corresponden a los implementados.

De la central se envían hacia todas las unidades el BUS de comunicaciones y dos cables de alimentación, uno de 24[V] dirigidas al cartel de entrada y a las URs y una línea de 220[V] que recorre los distintos lugares de estacionamiento, cuenta con dos pulsadores, los cuales simulan los sensores que deben estar ubicados al ingreso y egreso para detectar cuando un vehículo está presente.

La UC es la única que cuenta con la posibilidad de conectarse a un tomacorriente.



Figura 34: Cartel de Entrada.

En la entrada de la playa, al ingresar un vehículo, el conductor observará el cartel de la Figura 34 para una orientación rápida al poder ver el estado de los lugares y así seleccionar el camino más conveniente hacia uno disponible.

### Capítulo 3. Resultados

Los resultados obtenidos al realizar este proyecto fueron satisfactorios ya que se cumplió el objetivo principal de detectar patentes y los caracteres que las componen, conseguir un correcto sensado de los lugares para determinar si un vehículo se encuentra estacionado o no, y lograr comunicar todas las unidades incluyendo el cartel de forma inmediata y sin errores.

En cuanto al procesamiento de imágenes se encontraron ciertas limitaciones, algunas de ellas excediendo la posibilidad de la automatización como ser el caso de patentes en mal estado o la ausencia de las mismas y otras posibles de corregir mediante programación como ser eventuales casos de patentes que ingresan con una determinada perspectiva lo que imposibilita su correcta detección, aunque este caso sea ajeno a lo planteado en el prototipo del proyecto.

La resolución en la imagen de la captura del vehículo juega un papel importante para la posterior detección de la patente y sus caracteres. Por lo que al trabajar con cámaras IP nativas, se consiguió una mayor resolución ya que estas no tienen pérdidas por compresión y de esta manera resolver las exigencias de calidad de imagen.

La utilización del sensor HC-SR04 permitió una gran precisión al momento de detectar la presencia de un vehículo, debido a que es posible variar el alcance de detección por software y este permanece constante mientras que no se modifique su valor.

Al utilizar el protocolo RS-485 operando a una velocidad de 9600[baudios] se consiguió establecer una comunicación entre las distintas unidades y el cartel de entrada, pudiendo agregar en un futuro un mayor número de URs al BUS de comunicaciones.

La aplicación de PC cumplió con todas las tareas que debía llevar a cabo, sin embargo el tiempo de respuesta al momento de procesar las imágenes es mucho mayor a lo esperado, haciéndose más notable cuanto más cerca esté la patente de la cámara.

## Capítulo 4. Análisis de Costos

El siguiente análisis de costos se realizó para un estacionamiento de 60 lugares:

Material	Cantidad	Precio Unitario [\$]	Precio Total [\$]
Camara IP	2	1.800	3.600
Sensor ultrasónico	62	50	3.100
Gotas Led	120	30	3.600
Gabinete	61	80	4.880
Placa de UC	1	400	400
Cartel de entrada	1	3.000	3.000
Placa de UR	60	200	12.000
Gastos extras	1	4.500	4.500
Total			35.080

**Tabla 2: Análisis de costos de insumos.**

En el proyecto se invirtieron alrededor de 8 horas diarias, 6 días a la semana, en un período de 7 meses, dando un tiempo aproximado de 1344 horas. Tomando como referencia el Colegio de Ingenieros Especialistas de Entre Ríos (CIEER) en su Unidad Arancelaria, Artículo 5 de la Resolución N°1285/2016, que establece un monto de \$160 por hora de trabajo, se tendría un total de \$215.040. Considerando que el presente proyecto fue desarrollado por dos personas, el costo de diseño será de \$430.080. El cual no se volverá a tener cuando se implemente nuevamente el sistema. En los costos de Placa de UR y Placa de UC se incluyó el costo de fabricación de las mismas por parte de una empresa dedicada a esto.

El precio de venta del sistema total incluyendo instalación, aplicación de PC y ensamblado de unidades remotas será de \$120.000.

A partir de los datos anteriores, se puede encontrar la cantidad de sistemas que deben venderse para saldar el costo de diseño, la cual se encuentra obteniendo el cociente entre el costo de diseño y la ganancia por cada venta.

Al vender el 5° sistema se saldará el costo de diseño y cualquier venta posterior pasaría a ser ganancia.

## Capítulo 5. **Discusión y Conclusión**

El objetivo propuesto de detectar las patentes que circulan actualmente en el país (formato de 1995 y del Mercosur) y reconocer correctamente los caracteres que las componen se logró pero con ciertas limitaciones dado el estado de las mismas, la falta de tipografía de las patentes con formato de 1995 y las condiciones del ambiente que puedan dificultar la detección (luz, inclinación y perspectivas de profundidad).

El mal estado de muchas patentes obligó a que la aplicación de PC pueda brindarle al operario la posibilidad de ingresar la patente en caso de que el programa no la detecte o corregirla si el código identificado fuera erróneo.

La falta de una fuente tipográfica para los caracteres de las patentes con formato de 1995 se solucionó creando una fuente propia con recortes de imágenes tomadas a lo largo del desarrollo de este proyecto, generando así un abecedario y los 10 dígitos que deben ser cargados al OCR para que este pueda reconocer los caracteres en las capturas de las patentes.

Las condiciones del ambiente que puedan ser desfavorables para la detección de las patentes se corrigieron variando la configuración de imagen de la cámara IP, adaptándola a cada ambiente en particular (hay que tener en cuenta que la cámara debe ser instalada en un lugar fijo y con un ambiente con pocas variaciones como ser el interior de una playa de estacionamiento).

Así el sistema obtenido permite detectar patentes y reconocer sus caracteres a una distancia frontal entre 1 y 3[m]. Concordando así el desempeño obtenido con el presentado en la bibliografía de la cual se tomó de referencia el procesamiento digital de imágenes, adaptando este para las características de las patentes vigentes en nuestro país.

Como punto negativo cabe destacar que el tiempo requerido para el procesamiento digital de imágenes necesario para detectar las patentes y sus caracteres es mucho mayor al esperado a comienzos del proyecto.

La señalización en tiempo real del estado de disponibilidad de los lugares dentro de la playa de estacionamiento se logró mediante el uso del sensor HC-SR04 y la comunicación entre las unidades remotas y la central a través de un BUS de

comunicaciones sobre el cual se implementó el protocolo RS-485. Además, se consiguió reservar los lugares desde la aplicación de PC y ver los cambios de estado de los lugares casi de inmediato.

El sensor ultrasónico HC-SR04 tiene una gran precisión pero debe ser controlado en cada UR, y ésta a la vez debe esperar a que la UC le indique que debe sensar. No se puede utilizar el sensor libremente en cada UR porque esto produce que los mismos se interfieran al estar funcionando conjuntamente.

El BUS de comunicaciones funcionó correctamente pero se debieron tener en cuenta ciertas consideraciones sobre el transceptor con el que se conectan todas las unidades al canal de comunicaciones. Algunas de ellas no aparecen siquiera en la hoja de datos como ser el uso de una resistencia de pull-down en los pines que conmutan el funcionamiento del transceptor para que transmita o reciba, de manera que se llegó a esta solución empíricamente.

Como mejoras a futuro se pueden citar reemplazar el BUS de comunicaciones cableado y los transceptores MAX487 por módulos RF, evitando así el tendido de cables de comunicación. Además, se podría volver a utilizar las fuentes capacitivas en cada UR con lo que también se evitaría usar los cables de 24[V] del transformador, ahorrando así dinero en tendido de cables y en los tubos que los contengan.

Otra mejora sería diseñar una aplicación para teléfonos móviles de modo el cliente pueda observar en su celular en detalle la hora de ingreso, la de egreso, el tiempo transcurrido en el estacionamiento, la tarifa aplicada y el monto a cobrar, evitando así el uso de papel.

Por último, se podría diseñar una página web similar a la aplicación de PC en la que el encargado o dueño de la playa pueda acceder remotamente y observar la cantidad de lugares libres y ocupados, y todas las estadísticas definidas anteriormente en la aplicación pero sin la posibilidad de realizar ajustes, cobros ni reservas.

## Capítulo 6. Literatura Citada

1. Baggio, D., Escrivá, D., Mahmood, N., Shilkrot, R., Emami, S., Ievgen, K., Saragih, J., (2012), *Mastering OpenCV with Practical Computer Vision Projects*, Birmingham, Reino Unido: Packt Publishing Ltd.
2. Black Box (2005-2017). Black Box Network Services. México: Cámaras analógicas vs. Cámaras IP: Una comparativa en 12 puntos. Recuperado de <https://www.blackbox.com.mx/mx-mx/page/23769/Recursos/News-Events/News/cmaras-analgicas-vs-cmaras-ip-una-comparativa-en-12puntos>.
3. Método del valor umbral. (Sin fecha). En *Wikipedia*. Recuperado el 6 de noviembre de 2017 de [https://es.wikipedia.org/wiki/M%C3%A9todo\\_del\\_valor\\_umbral](https://es.wikipedia.org/wiki/M%C3%A9todo_del_valor_umbral).
4. Reston Condit y Microchip Inc.. (2004). Transformerless Power Supplies: Resistive and Capacitive. *AN954*, 1-14.

Universidad Tecnológica Nacional  
Facultad Regional Paraná

Se certifica que ..... , DNI: ..... ha realizado la  
dirección del Proyecto Final:

.....  
.....  
.....

De los alumnos:

- .....
- .....
- .....

Realizada durante el ciclo lectivo: ....., obteniendo el grupo un calificación final  
de: .....

A fin de ser emitida la correspondiente certificación por el departamento de electrónica, se  
extiende la siguiente constancia.

**Pañoni Sergio**

**Ramos Hector**

**Maggiolini Lucas**