



# **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECATRÓNICA**

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE MEDICIÓN DE VELOCIDAD PARA VEHICULOS UTILIZANDO VISIÓN ARTIFICIAL IMPLEMENTADO TECNOLOGÍA RASPBERRY PI.”**

## **AUTORES:**

**Marco Vinicio De La Cruz Guerrero  
Paul Vicente Borja Guerra**

## **DIRECTOR:**

**Ing. Eddie Galarza  
2017**

# INTRODUCCIÓN

- La visión artificial se la puede definir como un anexo que envuelve a la especialidad de “inteligencia artificial”; la cual por medio de utilización de técnicas adecuadas, permite la adquisición, procesamiento y estudio de cualquier tipo de información a través de imágenes digitales.



# AGENDA

objetivos e hipotesis

Descripción del sistema y sus componentes

Algoritmo

Desarrollo del proyecto

Análisis de resultados

Conclusiones y Recomendaciones



# Objetivo General

Diseñar y construir un sistema de medición de velocidad para vehículos con visión artificial.



# Objetivos Específicos.

Recopilar información sobre los componentes que conforman un radar de velocidad.

Diseñar e implementar el sistema mecánico y electrónico utilizando material en el mercado nacional

Realizar pruebas de campo del sistema de medición de velocidad.

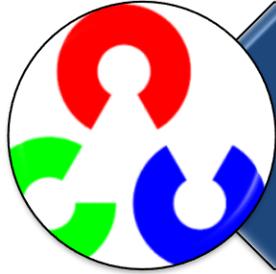


# HIPÓTESIS

¿El Diseño y construcción de un sistema de medición de velocidad de vehículos nos permitirá medir la velocidad a la que circulan los vehículos en las vías públicas?



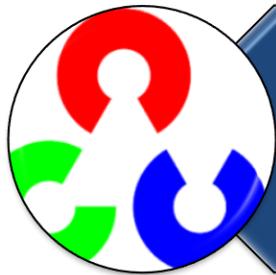
# OPENCV



Librería de software libre que se utiliza para visión artificial de manera académica como comercial



OpenCv es multiplataforma ya que va teniendo versiones para diferentes sistemas operativos



Está integrado con interfaces de C++, C, Python y últimamente con Java, para aplicaciones en tiempo real.



# TARJETAS EMBEBIDAS

Se encuentran presentes en casi todos los aparatos utilizados en nuestra vida cotidiana.

La mayoría de sistemas enbebido se logran programar con el uso específico de compiladores como lenguaje C, C ++, o directamente con el lenguaje ensamblador del chip.

Sistema electrónico fundamentalmente diseñado para funciones específicas en tiempo real con pocos recursos.

Un sistema en bebido tiene una arquitectura semejante a una pc convencional.

Existen varios tipos de tarjetas como: Arduino ,rabbit, BeagleBone, Raspberry Pi.



# TARJETA RASPBERRY PI



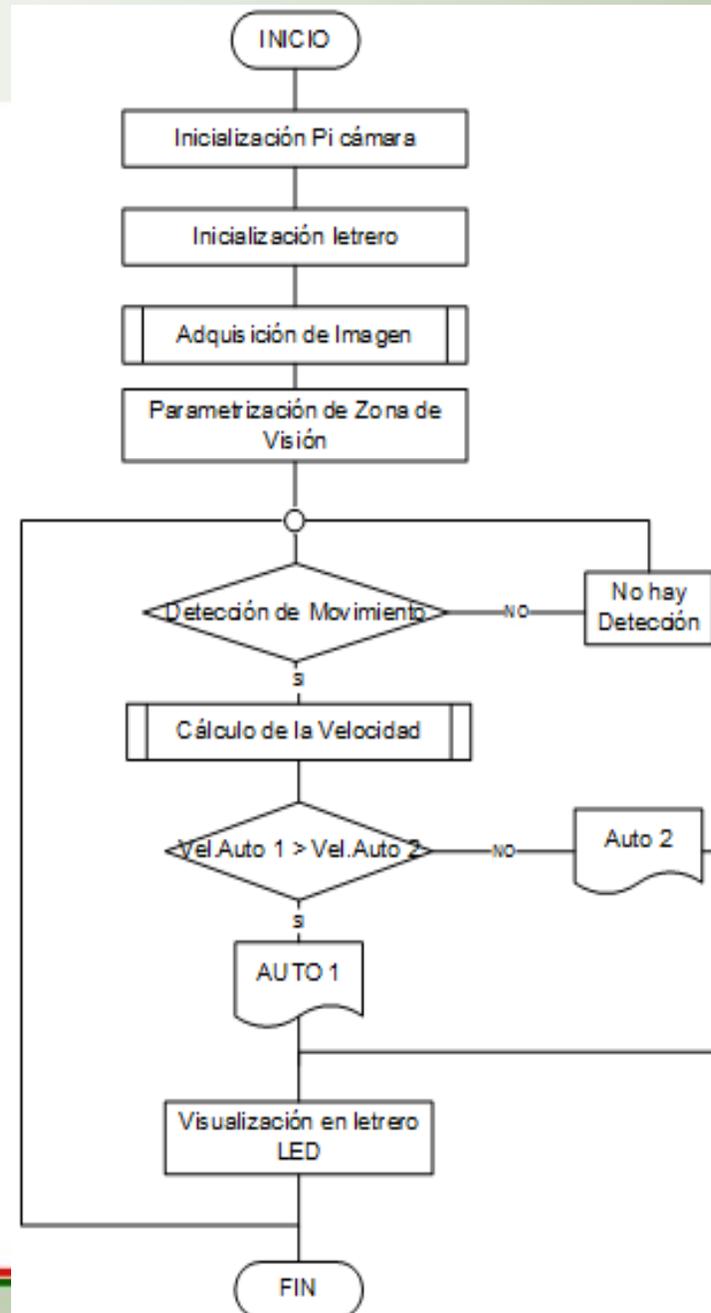
**ESPE**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA



- Microprocesador de 900Hz
- 1GB de RAM
- Utiliza la plataforma Raspbian



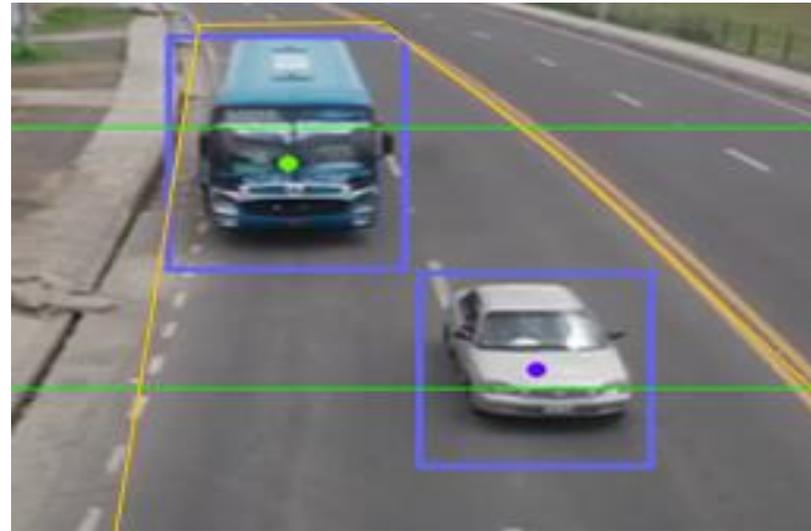
# Algoritmo



- Inicialización de pi cámara.
- Inicialización de letrero led.
- Adquisición de imagen.



La adquisición de las imágenes en video de entrada se realiza mediante la cámara propia de la tarjeta Raspberry pi la cual toma la secuencia de imágenes y procede al respectivo análisis mediante el algoritmo

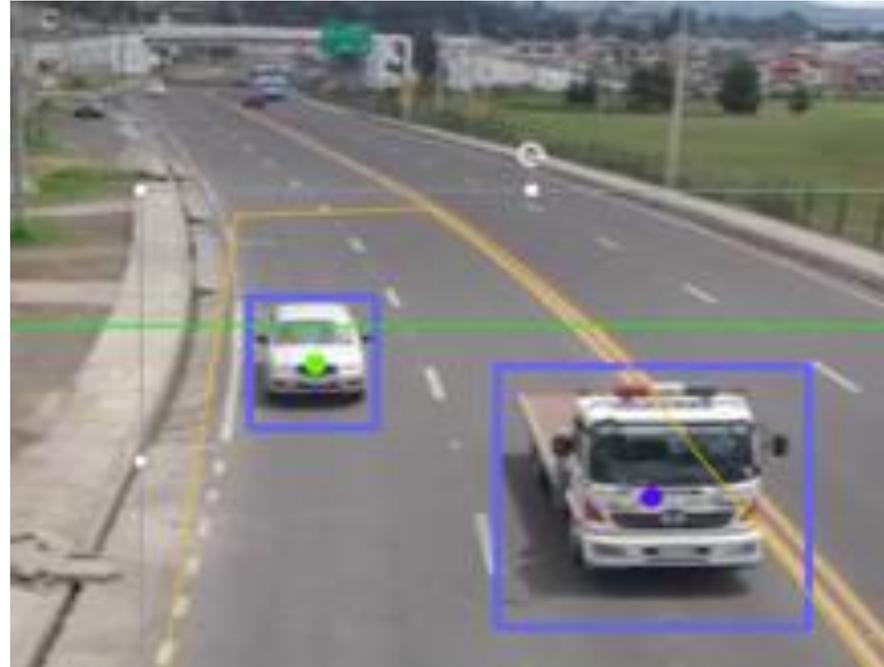


Secuencia de imágenes capturas mediante la pi cámara

- Parametrización de zona de visión
- Detección de movimiento.
- Cálculo de velocidad.



En el algoritmo se realizó la parametrización para analizar una sección específica

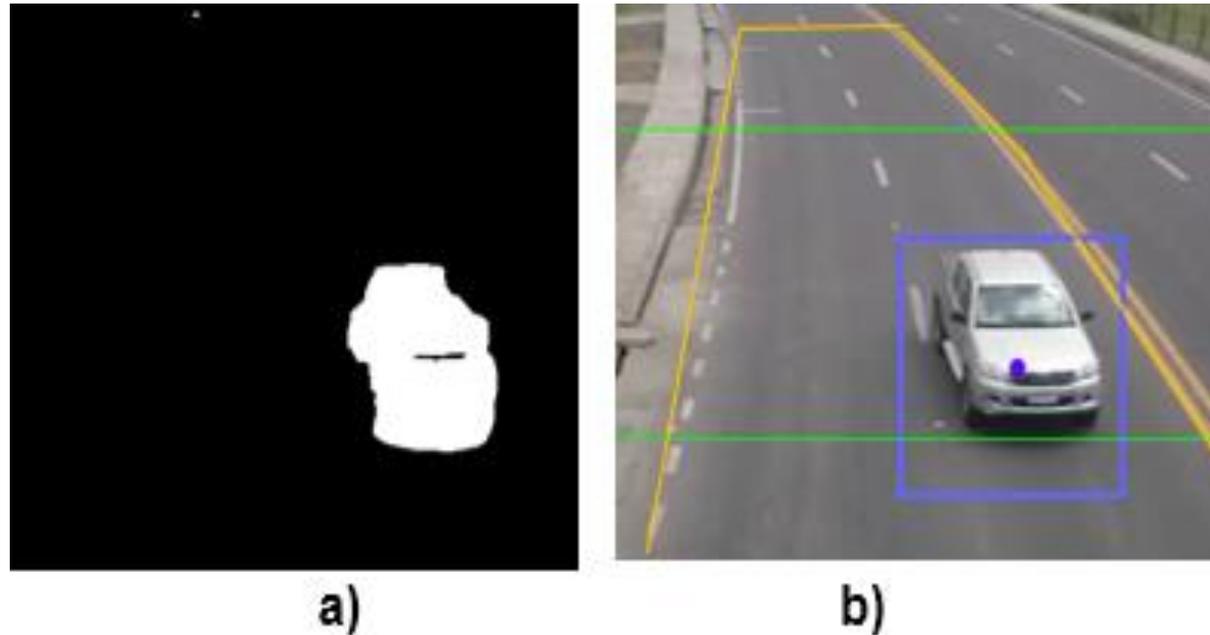


La línea amarilla muestra la parametrización

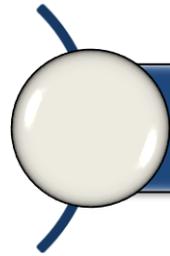


**E S P E**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA

En la detección de movimiento se observa en la imagen a) el escalamiento en escala de grises para facilitar la detección de vehículos, en la imagen b) se observa el cuadro azul que determina el vehículo cuya velocidad será calculada



Detección de Autos



## Visualización de números

Para visualizar números en la matriz de leds se implementó un algoritmo en Arduino para el cual se utilizó la librería SPI (Serial Peripheral Interface) que es un protocolo de datos en serie síncrono, propio del Arduino para la comunicación con dispositivos periféricos, permitiendo una mayor velocidad en distancias cortas



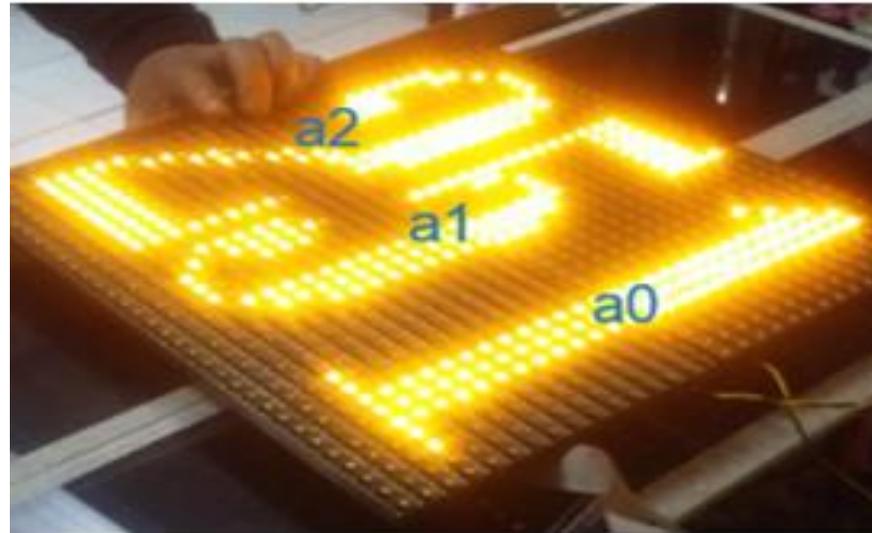
La librería DMD.h (Dot Matriz Display) es una biblioteca para el manejo de matrices de Leds de un diseño de 32 x 16, la cual se obtuvo de la web (<https://github.com/freetronics/DMD>), la misma permite controlar directamente las matrices de leds con comandos pre-programados

```
void modificarV() {  
    a0 = int((b%10)+48);  
    a1 = int(((b%100)/10)+48);  
    a2 = int(((b%1000)/100)+48);  
}
```

**Especificación de orden en la matriz**



Se van presentando los números de derecha a izquierda en las posiciones especificadas, donde a0 corresponde al primer dígito, a1 al segundo y a2 al tercero



# Análisis de resultados

- Las pruebas de campo que se realizaron fueron en base a las siguientes dos variables:
- Análisis de detección.
- Análisis de tiempos de procesamiento en detección.
- Análisis de rendimiento de tarjeta y software.

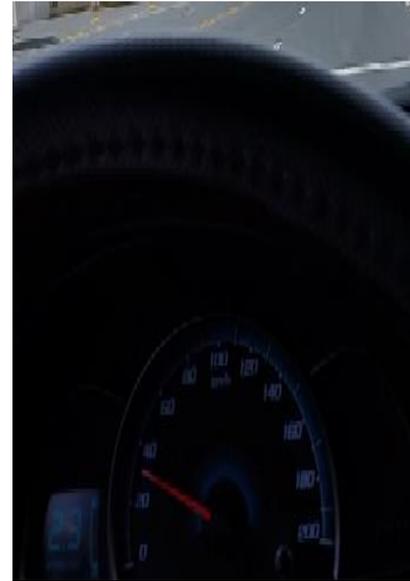


# Rendimiento de detección de vehículos.

- En las pruebas experimentales se realizó la detección de vehículos por imágenes en video por medio de Open CV y la plataforma de raspbian



**Detección del vehículo a 34km**



**Marcación de velocímetro de auto 37km**

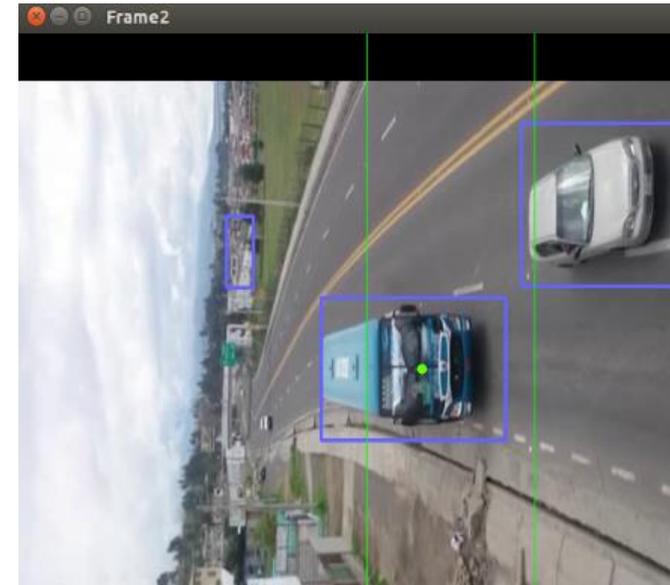


# Rendimiento de detección y procesado

Se realizó varios ensayos del algoritmo para verificar su funcionalidad y confiabilidad del mismo además se observó que el tiempo de detección varía cuando existe más cantidad de vehículos en la zona de parametrización.

```
markino@markino-Dell-System-Inspiron-N
AUTO 1
AUTO 2
VELOCIDAD AUTO 1: 0.0 Km/h
AUTO 1
VELOCIDAD AUTO 1: 34.53 Km/h
VELOCIDAD AUTO 2: 33.13 Km/h
AUTO 1
AUTO 2
VELOCIDAD AUTO 1: 0.0 Km/h
VELOCIDAD AUTO 2: 0.0 Km/h
AUTO 1
VELOCIDAD AUTO 1: 55.15 Km/h
AUTO 1
AUTO 2
VELOCIDAD AUTO 1: 49.25 Km/h
VELOCIDAD AUTO 2: 21.8 Km/h
AUTO 1
VELOCIDAD AUTO 1: 30.71 Km/h
AUTO 1
VELOCIDAD AUTO 1: 37.85 Km/h
AUTO 1
AUTO 2
VELOCIDAD AUTO 1: 43.44 Km/h
```

Detecciones de vehículos



Detección de vehículos en zona de parametrización



# Resultados de detección

Para la detección de velocidad se trazó una zona de visión o parametrización, esta forma un polígono en la carretera permitiendo visualizar los vehículos están movimiento.



zona de parametrizada

# HIPÓTESIS PLANTEADA Y CUMPLIMIENTO DE LA MISMA

- Con el uso de plataformas de software libre en nuestro caso Python y la tarjeta Raspberry Pi se realizó varias pruebas de campo del sistema de medición de velocidad realizadas y con estas se demostró que el proyecto es factible.

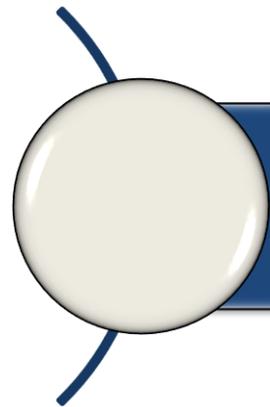
## **Los aspectos principales que establecen el cumplimiento de la hipótesis son los siguientes:**

- La tarjeta Raspberry PI cuenta con su propio sistema operativo Raspbian el cual está basado en Linux y es una plataforma de software libre, también cabe recalcar que la tarjeta es muy económica y de fácil adquisición.



- La Pi cámara facilita la implementación del algoritmo ya que la misma brinda una velocidad de transferencia ideal para que trabaje el sistema con el menor margen de error en la adquisición.
- Las características técnicas de la Raspberry Pi permite trabajar en el procesamiento de imágenes bajo la plataforma de Python para el desarrollo del sistema de detección de velocidad vehicular.





# CONCLUSIONES Y RECOMENDACIONES



# CONCLUSIONES

- El sistema de detección de velocidad vehicular está basado en un algoritmo realizado con OpenCV bajo la plataforma Python el mismo permite un desarrollo eficiente, sencillo y de bajo costo en comparación de otros lenguajes de programación.
- El cambio constante en la luminosidad da lugar a que se produzcan detecciones y mediciones erróneas, no obstante, la tasa de error en este aspecto es lo suficientemente pequeña entre 5 km/h lo cual se encuentra dentro de los parámetros aceptables de error.
- Para disminuir los errores por el cambio de luminosidad se debe realizar un mayor nivel de procesamiento de las imágenes con lo cual aumenta el costo computacional y encarecería el proyecto.
- La eficacia del sistema de alimentación disminuye cuando existe menor intensidad lumínica del sol ya que se reduce la aportación del panel solar al sistema de baterías.



# RECOMENDACIONES

- Es necesario estudiar previamente la plataforma de programación Python y su integración con OpenCV.
- Instalar correctamente todos los paquetes de OpenCV necesarios para su correcto funcionamiento con Python.
- Estudiar todo lo referente a conexiones y funcionamiento de la tarjeta Raspberry Pi 2 y su Pi camera para evitar daños por mala conexión y evitar su mal funcionamiento.
- Aislar las baterías de la superficie metálicas para evitar su descarga por contacto.
- Verificar que la Pi cámara funcione correctamente con OpenCV realizar pruebas de funcionamiento.

