

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Algoritmos de Visión por computador para el  
reconocimiento de la pose de agentes empleando programación  
orientada a objetos y multi-hilos**

Trabajo de graduación presentado por José Pablo Guerra Jordán para  
optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2020







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Algoritmos de Visión por computador para el  
reconocimiento de la pose de agentes empleando programación  
orientada a objetos y multi-hilos**

Trabajo de graduación presentado por José Pablo Guerra Jordán para  
optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2020



Vo.Bo.:

(f) \_\_\_\_\_  
Ing. Luis Rivera

Tribunal Examinador:

(f) \_\_\_\_\_  
Ing. Luis Rivera

(f) \_\_\_\_\_  
XXX

(f) \_\_\_\_\_  
XXX

Fecha de aprobación: Guatemala, DIA X de MES XXX de 2020.





Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae eleifend ipsum, ut mattis nunc. Pellentesque ac hendrerit lacus. Cras sollicitudin eget sem nec luctus. Vivamus aliquet lorem id elit venenatis pellentesque. Nam id orci iaculis, rutrum ipsum vel, porttitor magna. Etiam molestie vel elit sed suscipit. Proin dui risus, scelerisque porttitor cursus ac, tempor eget turpis. Aliquam ultricies congue ligula ac ornare. Duis id purus eu ex pharetra feugiat. Vivamus ac orci arcu. Nulla id diam quis erat rhoncus hendrerit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate, metus vel efficitur fringilla, orci ex ultricies augue, sit amet rhoncus ex purus ut massa. Nam pharetra ipsum consequat est blandit, sed commodo nunc scelerisque. Maecenas ut suscipit libero. Sed vel euismod tellus.

Proin elit tellus, finibus et metus et, vestibulum ullamcorper est. Nulla viverra nisl id libero sodales, a porttitor est congue. Maecenas semper, felis ut rhoncus cursus, leo magna convallis ligula, at vehicula neque quam at ipsum. Integer commodo mattis eros sit amet tristique. Cras eu maximus arcu. Morbi condimentum dignissim enim non hendrerit. Sed molestie erat sit amet porttitor sagittis. Maecenas porttitor tincidunt erat, ac lacinia lacus sodales faucibus. Integer nec laoreet massa. Proin a arcu lorem. Donec at tincidunt arcu, et sodales neque. Morbi rhoncus, ligula porta lobortis faucibus, magna diam aliquet felis, nec ultrices metus turpis et libero. Integer efficitur erat dolor, quis iaculis metus dignissim eu.



<b>Prefacio</b>	<b>V</b>
<b>Lista de figuras</b>	<b>IX</b>
<b>Lista de cuadros</b>	<b>XI</b>
<b>Resumen</b>	<b>XIII</b>
<b>Abstract</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
<b>3. Justificación</b>	<b>5</b>
<b>4. Objetivos</b>	<b>7</b>
<b>5. Alcance</b>	<b>9</b>
<b>6. Marco teórico</b>	<b>11</b>
<b>7. Resultados</b>	<b>15</b>
<b>8. Conclusiones</b>	<b>21</b>
<b>9. Recomendaciones</b>	<b>23</b>
<b>10. Bibliografía</b>	<b>25</b>
<b>11. Anexos</b>	<b>27</b>
11.1. Planos de construcción . . . . .	27
<b>12. Glosario</b>	<b>29</b>



---

## Lista de figuras

---

1.	Captura del primer archivo de texto para el ejemplo Multi-hilos . . . . .	15
2.	Captura del segundo archivo de texto para el ejemplo Multi-hilos . . . . .	16
3.	Captura del texto reconstruido para el ejemplo Multi-hilos . . . . .	16
4.	Captura de la cámara web con OpenCV . . . . .	17
5.	Prototipo de mesa 1 . . . . .	17
6.	Prototipo de mesa 2 . . . . .	18
7.	Prototipo de mesa 3 . . . . .	18
8.	Prueba calibración de la cámara . . . . .	19
9.	Pruebas calibración de la cámara 2 . . . . .	19
10.	Calibración de la cámara para detección de los códigos . . . . .	20
11.	Prueba de generación de código . . . . .	20



---

## Lista de cuadros

---

1.	Pruebas preliminares . . . . .	5
----	--------------------------------	---





Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae eleifend ipsum, ut mattis nunc. Pellentesque ac hendrerit lacus. Cras sollicitudin eget sem nec luctus. Vivamus aliquet lorem id elit venenatis pellentesque. Nam id orci iaculis, rutrum ipsum vel, porttitor magna. Etiam molestie vel elit sed suscipit. Proin dui risus, scelerisque porttitor cursus ac, tempor eget turpis. Aliquam ultricies congue ligula ac ornare. Duis id purus eu ex pharetra feugiat. Vivamus ac orci arcu. Nulla id diam quis erat rhoncus hendrerit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate, metus vel efficitur fringilla, orci ex ultricies augue, sit amet rhoncus ex purus ut massa. Nam pharetra ipsum consequat est blandit, sed commodo nunc scelerisque. Maecenas ut suscipit libero. Sed vel euismod tellus.

Proin elit tellus, finibus et metus et, vestibulum ullamcorper est. Nulla viverra nisl id libero sodales, a porttitor est congue. Maecenas semper, felis ut rhoncus cursus, leo magna convallis ligula, at vehicula neque quam at ipsum. Integer commodo mattis eros sit amet tristique. Cras eu maximus arcu. Morbi condimentum dignissim enim non hendrerit. Sed molestie erat sit amet porttitor sagittis. Maecenas porttitor tincidunt erat, ac lacinia lacus sodales faucibus. Integer nec laoreet massa. Proin a arcu lorem. Donec at tincidunt arcu, et sodales neque. Morbi rhoncus, ligula porta lobortis faucibus, magna diam aliquet felis, nec ultrices metus turpis et libero. Integer efficitur erat dolor, quis iaculis metus dignissim eu.



---

## Abstract

---

This is an abstract of the study developed under the



# CAPÍTULO 1

---

## Introducción

---

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque eget consequat risus. Praesent a quam lacinia, consequat eros id, auctor tellus. Phasellus a dapibus arcu, vitae luctus leo. Aliquam erat volutpat. Suspendisse ac velit quam. Nullam risus nibh, lobortis vehicula elit non, pellentesque volutpat odio. Donec feugiat porta sapien gravida interdum. Cras odio nunc, lobortis sed pellentesque imperdiet, facilisis eu quam. Praesent pharetra, orci at tincidunt lacinia, neque nulla ornare lacus, ut malesuada elit risus non mi. Fusce pellentesque vitae sapien sed mollis. Curabitur viverra at nulla vitae porta. In et mauris lorem.

Vestibulum faucibus fringilla justo, eget facilisis elit convallis sit amet. Morbi nisi metus, hendrerit quis pellentesque non, faucibus at leo. Proin consectetur, est vel facilisis facilisis, arcu felis vestibulum quam, et fringilla metus neque at enim. Nunc justo mauris, egestas quis maximus eget, viverra vehicula nunc. Fusce eu nulla elementum, condimentum diam at, aliquam leo. Nullam sed sodales enim, eu imperdiet risus. Aliquam ornare augue leo, fringilla mattis nunc facilisis eget. Nam faucibus, libero a aliquet fermentum, magna arcu ultrices lacus, a placerat tortor turpis ut purus.

Integer eget ligula non metus egestas rutrum sit amet ut tellus. Aliquam vel convallis est, eu sodales leo. Proin consequat nisi at nunc malesuada gravida. Aliquam erat volutpat. Aliquam finibus interdum dignissim. Etiam feugiat hendrerit nisl, hendrerit feugiat ex malesuada in. Cras tempus eget arcu vitae congue. Ut non tristique mauris. Vivamus in mattis ipsum. Cras bibendum, enim bibendum commodo accumsan, ligula nulla porttitor ex, et pharetra eros nisl eget ex. Morbi at semper arcu. Curabitur massa sem, maximus id metus ut, molestie tempus quam. Vivamus dictum nunc vitae elit malesuada convallis. Donec ac semper turpis, non scelerisque justo. In congue risus id vulputate gravida. Nam ut mattis sapien.



## CAPÍTULO 2

---

### Antecedentes

---

Puede encontrarse un trabajo similar en [1] o bien [2]





hgjhjjhvjvhgvjhgvjhg

12	3.2	3.43	23	13
aasdasdd	asd	ssdssa	ssdas	asdasda

Cuadro 1: Pruebas preliminares. Este cuadro corresponde a las pruebas realizadas durante blabla



#### **Objetivo General**

Mejorar el algoritmo de visión por computadora desarrollado para la mesa de pruebas Robotat para experimentación de robótica de enjambre, usando programación orientada a objetos, la librería OpenCV, y programación multi-hilos.

#### **Objetivos Específicos**

- Seleccionar el lenguaje de programación orientado a objetos más adecuado para la implementación de algoritmos de visión por computador para la mesa de pruebas Robotat.
- Desarrollar algoritmos computacionalmente eficientes por medio de programación multi-hilos.
- Diseñar e implementar una herramienta de software para aplicaciones de robótica de enjambre, usando los algoritmos desarrollados.
- Validar la herramienta de software en la mesa de pruebas Robotat.



## CAPÍTULO 5

---

Alcance

---

Podemos usar Latex para escribir de forma ordenada una fórmula matemática.



### Visión por Computadora

Visión por computadora se refiere al uso de cámaras o cualquier otro dispositivo de toma de fotografías o vídeos, para recolectar información para su posterior análisis, desarrollando algoritmos para hacer entender a la computadora que es lo que hay (en cuanto a datos o información significativa) en este tipo de archivos. [3]

En otras palabras, consiste en obtener la información relevante, realizando un procesamiento a imágenes y vídeos, para que los seres humanos puedan entender de mejor manera que es lo que hay en ellos. Es decir, poder visualizar lo que una computadora hace en este tipo de procesamientos. Normalmente, este tipo de procesamiento de datos es utilizado para obtener información del medio o entorno (mapas, carreteras, imágenes de todo tipo, etc.) y poder ser utilizado en resolución de problemas o toma de decisiones por parte de una computadora, basado en su entorno o aplicación [3]

### OpenCV

Esta es la librería de **Open Source Computer Vision Library** y está disponible para Windows, MacOS y Linux. Es una librería para visión por computadora y machine learning. Incluye mas de 2500 algoritmos que permiten ser utilizados para reconocimiento de rostros, identificación de objetos, clasificación del comportamiento humano, rastreo del movimiento de los ojos entre otros.

Su implementación puede realizarse en C++, Python, Java y Matlab para las diferentes aplicaciones mencionadas. [4]

## Programación Multi-hilos

Los procesadores orientados a multiprocesos, permiten realizar diferentes tareas al mismo tiempo. Estos a su vez, son responsables de manejar los recursos que se le asignen a cada uno de estos.

Más específicamente, cuando un programa es ejecutado, la computadora crea algo llamado **proceso** que contiene toda la información relevante del programa, como su identificador por ejemplo, hasta que dichos los programas terminan su ejecución.

Los sistemas operativos multiprocesos, permiten la ejecución de varios programas de manera simultánea y es la computadora la encargada de asignar los recursos a los diferentes programas que los necesiten. El objetivo principal es obtener un uso adecuado de los recursos del CPU. [5]

Un hilo, por tanto, es una unidad de control dentro de un proceso. El programa corre un hilo principal o **main thread** encargado de crear otros hilos según sea su programación, siendo este el principio básico del multi-hilos. Básicamente, orientar los programas ejecutados para realizar varias tareas y en muchos casos realizar procesos más eficientes. [5]

Existen ventajas en el uso de multi-hilos. El primero, es la ventaja de tener un programa realizando captura o procesamiento de información, mientras otro está esperando estas salidas. Es decir, en lugar de tener un programa que espera una imagen o dato, para luego procesarla y dar un resultado, se pueden tener dos procesos corriendo, uno capturando datos y el otro procesando. El resultado de esto será mucho más rápido.

Además, mencionar que la comunicación multi-hilos es mucho más eficiente que la comunicación entre procesos [5]

## Programación Orientada a Objetos

Es un enfoque para la organización y el desarrollo de programas que intenta incorporar características más potentes y estructuradas para la programación. Es una nueva forma de organizar y desarrollar programas y no tiene nada que ver con ningún lenguaje de programación específico. Sin embargo, no todos los lenguajes son adecuados para implementar fácilmente los conceptos de la Programación orientada a objetos (POO). [6]

### Objetos

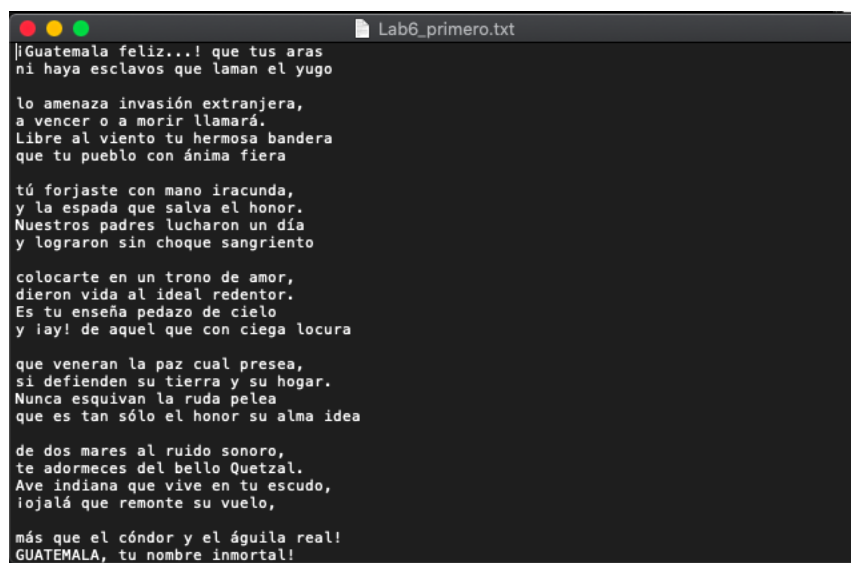
Un objeto en POO cumple básicamente las mismas funciones que un objeto de la vida real. Un objeto puede guardar atributos o características y ser utilizadas mediante métodos. Esto es útil porque permite de alguna forma esconder esta parte interna para solo enfocarse en su función principal (aquella para la cual fueron diseñados). A esto se le conoce como *encapsulación*. [7]



## Clase

Una clase es el plano a partir del cual se crean métodos o atributos para los diferentes objetos individuales que pertenecen a esta clase. Es, lo que se podría llamar, como un molde o modelo para la creación de objetos. [8] [7]



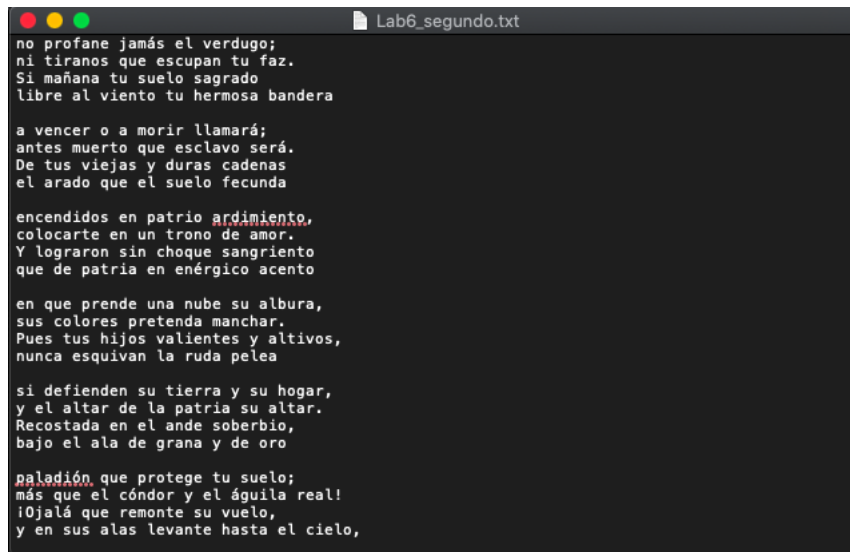


```
||Guatemala feliz...! que tus aras  
ni haya esclavos que laman el yugo  
  
lo amenaza invasión extranjera,  
a vencer o a morir llamará.  
Libre al viento tu hermosa bandera  
que tu pueblo con ánima fiera  
  
tú forjaste con mano iracunda,  
y la espada que salva el honor.  
Nuestros padres lucharon un día  
y lograron sin choque sangriento  
  
colocarte en un trono de amor,  
dieron vida al ideal redentor.  
Es tu enseña pedazo de cielo  
y ¡ay! de aquel que con ciega locura  
  
que veneran la paz cual presea,  
si defienden su tierra y su hogar.  
Nunca esquivan la ruda pelea  
que es tan sólo el honor su alma idea  
  
de dos mares al ruido sonoro,  
te adormeces del bello Quetzal.  
Ave indiana que vive en tu escudo,  
¡ojalá que remonte su vuelo,  
  
más que el cóndor y el águila real!  
GUATEMALA, tu nombre inmortal!
```

Figura 1: Captura del primer archivo de texto para el ejemplo Multi-hilos

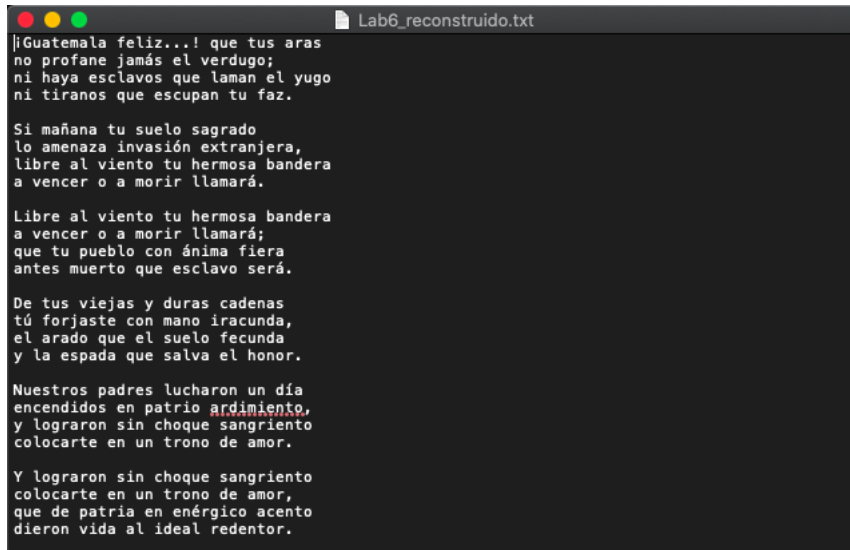
Las figuras anteriores detallan 3 archivos de texto. Estos archivos son utilizados para el programa ejemplo/prueba de programación multi-hilos. El objetivo principal era reconstruir el himno nacional de Guatemala (dividido en dos archivos diferentes) y ese resultado guardarlo en un tercer archivo de texto.

Se emplea la programación multi-hilos ya que es un buen ejemplo de demostrar como se pueden tener varios hilos corriendo tareas diferentes, accediendo a un mismo recurso y obtener un resultado de manera más eficiente.



```
no profane jamás el verdugo;  
ni tiranos que escupan tu faz.  
Si mañana tu suelo sagrado  
libre al viento tu hermosa bandera  
  
a vencer o a morir llamará;  
antes muerto que esclavo será.  
De tus viejas y duras cadenas  
el arado que el suelo fecunda  
  
encendidos en patrio ardimiento,  
colocarte en un trono de amor.  
Y lograron sin choque sangriento  
que de patria en enérgico acento  
  
en que prende una nube su albura,  
sus colores pretenda manchar.  
Pues tus hijos valientes y altivos,  
nunca esquivan la ruda pelea  
  
si defienden su tierra y su hogar,  
y el altar de la patria su altar.  
Recostada en el ande soberbio,  
bajo el ala de grana y de oro  
  
paladín que protege tu suelo;  
más que el cóndor y el águila real!  
¡Ojalá que remonte su vuelo,  
y en sus alas levante hasta el cielo,
```

Figura 2: Captura del segundo archivo de texto para el ejemplo Multi-hilos



```
¡Guatemala feliz...! que tus aras  
no profane jamás el verdugo;  
ni haya esclavos que laman el yugo  
ni tiranos que escupan tu faz.  
  
Si mañana tu suelo sagrado  
lo amenaza invasión extranjera,  
libre al viento tu hermosa bandera  
a vencer o a morir llamará.  
  
Libre al viento tu hermosa bandera  
a vencer o a morir llamará;  
que tu pueblo con ánimo fiera  
antes muerto que esclavo será.  
  
De tus viejas y duras cadenas  
tú forjaste con mano iracunda,  
el arado que el suelo fecunda  
y la espada que salva el honor.  
  
Nuestros padres lucharon un día  
encendidos en patrio ardimiento,  
y lograron sin choque sangriento  
colocarte en un trono de amor.  
  
Y lograron sin choque sangriento  
colocarte en un trono de amor,  
que de patria en enérgico acento  
dieron vida al ideal redentor.
```

Figura 3: Captura del texto reconstruido para el ejemplo Multi-hilos



Figura 4: Captura de la cámara web con OpenCV

La figura anterior muestra una captura de video de la cámara web utilizando OpenCV y Python. El objetivo de esta demostración no es solo ver como funciona la librería y obtener una imagen, sino que también pasarle un filtro o procesamiento para obtener un resultado en escala de grises.



Figura 5: Prototipo de mesa 1

Finalmente, estas figuras muestran el primer prototipo de las mesas de pruebas con la que se estará trabajando. El objetivo es simular a escala la mesa que se encuentra en el laboratorio de la UVG. Como se puede observar tiene pequeñas marcas que servirán de referencia para la calibración al momento de las pruebas. Este modelo de mesa se buscará mejorar para tener una mejor posición de la cámara cuando las pruebas finales den inicio.



Figura 6: Prototipo de mesa 2



Figura 7: Prototipo de mesa 3

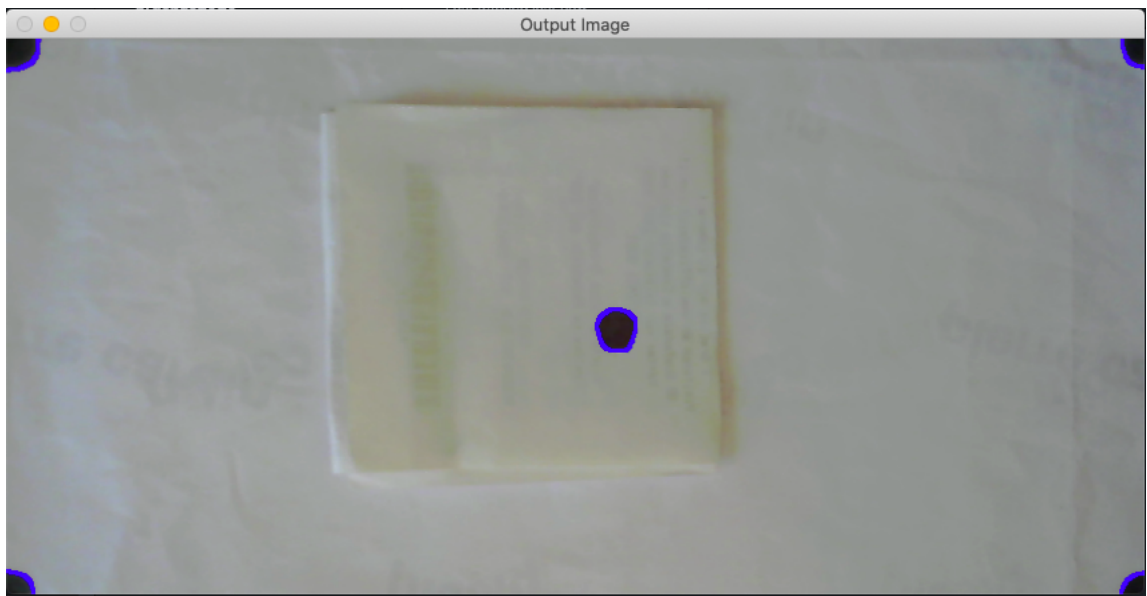


Figura 8: Prueba calibración de la cámara

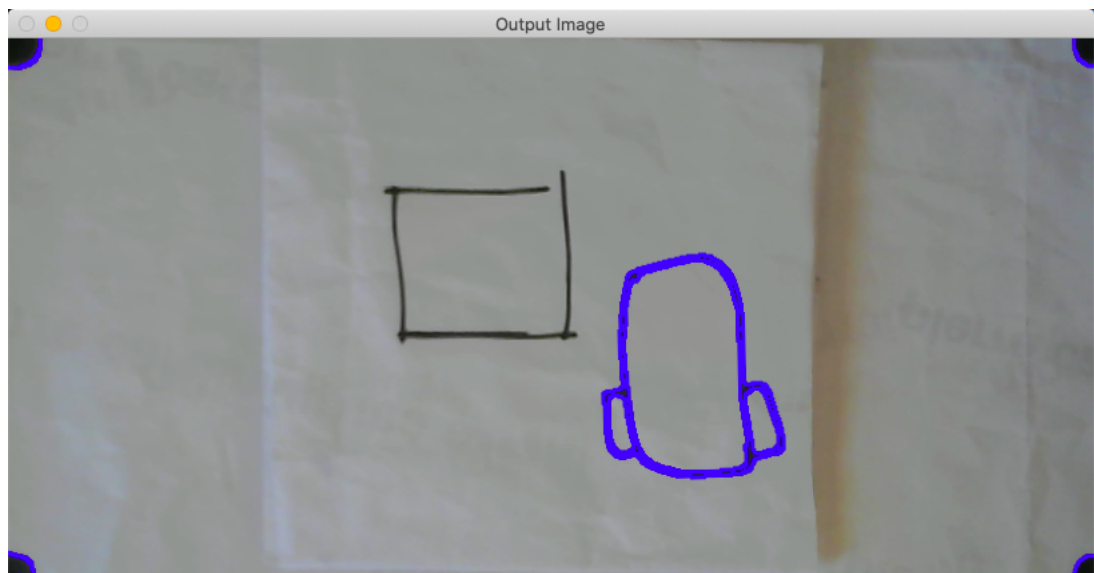


Figura 9: Pruebas calibración de la cámara 2

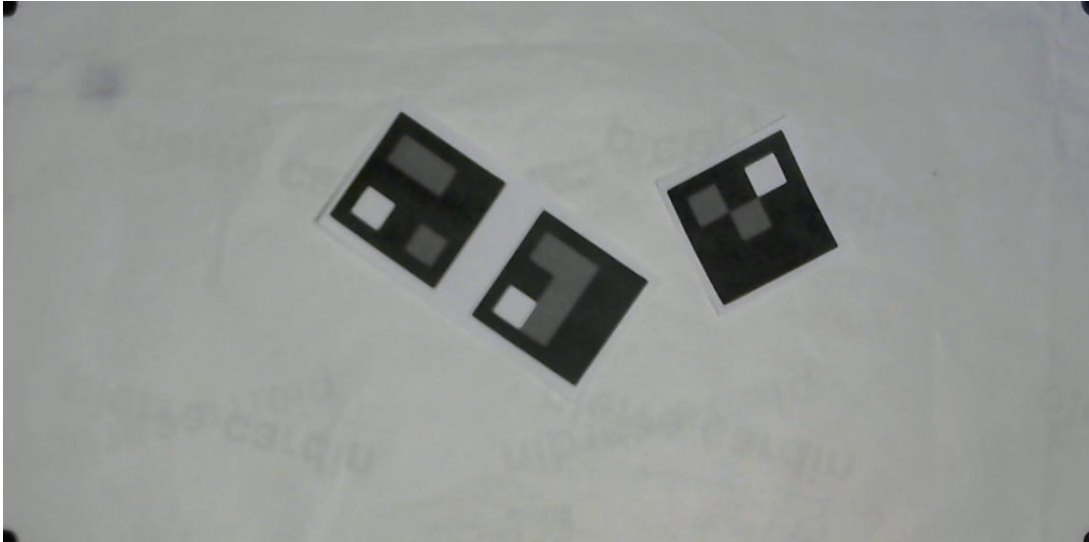


Figura 10: Calibración de la cámara para detección de los códigos

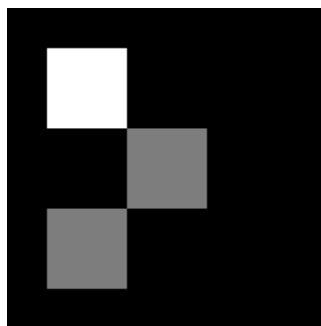


Figura 11: Prueba de generación de código



## CAPÍTULO 8

---

Conclusiones

---



## CAPÍTULO 9

---

Recomendaciones

---



- [1] A. M. Hoover, S. Burden, X.-Y. Fu, S. S. Sastry y R. S. Fearing, “Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot”, en *Biomedical Robotics and Biomechatronics (BioRob)*, 2010 3rd IEEE RAS and EMBS International Conference on, IEEE, 2010, págs. 869-876.
- [2] Y.-L. Park, B.-r. Chen, N. O. Pérez-Arancibia, D. Young, L. Stirling, R. J. Wood, E. C. Goldfield y R. Nagpal, “Design and control of a bio-inspired soft wearable robotic device for ankle-foot rehabilitation”, *Bioinspiration & biomimetics*, vol. 9, n.º 1, pág. 016 007, 2014.
- [3] Raúl E. López Briega, *Visión por computadora*, <https://iaarbook.github.io/vision-por-computadora/>, visitado el 10/05/2020.
- [4] OpenCV, *About*, <https://opencv.org/about/>, visitado el 05/04/2020.
- [5] R. H. Carver y K.-C. Tai, *Modern multithreading: implementing, testing, and debugging multithreaded Java and C++/Pthreads/Win32 programs*. John Wiley & Sons, 2005.
- [6] P. Chawla, *OOP with C++*, <http://www.ddegjust.ac.in/studymaterial/mca-3/ms-17.pdf>, visitado el 05/06/2020.
- [7] Oracle, *Lesson: Object-Oriented Programming Concepts*, <https://docs.oracle.com/javase/tutorial/java/concepts/index.html>, visitado el 05/06/2020.
- [8] E. Doherty, *What is Object Oriented Programming? OOP Explained in Depth*, [https://www.educative.io/blog/object-oriented-programming?aid=5082902844932096&utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=blog-dynamic&gclid=Cj0KCQjwoPL2BRDxARIsAEMm9y\\_jZ7Mu3oH1wRJc5uHdWIeMdhGHbLeR22kkNxokV1-YZS-isYFjZKIaArzfEALw\\_wcB](https://www.educative.io/blog/object-oriented-programming?aid=5082902844932096&utm_source=google&utm_medium=cpc&utm_campaign=blog-dynamic&gclid=Cj0KCQjwoPL2BRDxARIsAEMm9y_jZ7Mu3oH1wRJc5uHdWIeMdhGHbLeR22kkNxokV1-YZS-isYFjZKIaArzfEALw_wcB), visitado el 05/06/2020.



## CAPÍTULO 11

---

Anexos

---

### 11.1. Planos de construcción





## CAPÍTULO 12

---

### Glosario

---

**fórmula** Una expresión matemática. 9

**latex** Es un lenguaje de marcado adecuado especialmente para la creación de documentos científicos. 9