



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Diseño e implementación de un sistema de control de calidad mediante visión por computador**

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Lorenzo Del Pianta Pérez

**Tutor:** Alberto José Pérez Jiménez

**Curso:** 2019/2020



---

## RESUMEN

Hoy en día, la calidad del producto es uno de los factores clave a la hora de diferenciarse de los competidores y, poder así obtener una ventaja competitiva. Las nuevas tecnologías ofrecen herramientas y métodos para poder asegurar altos estándares de calidad, no obstante, a veces las pequeñas y medianas empresas no poseen los recursos necesarios para implantar estas tecnologías debido a su alto coste.

Este trabajo consiste en el desarrollo de una herramienta con un bajo coste de implantación, además de una alta escalabilidad, que permita reconocer los posibles defectos de los productos a lo largo de su cadena de producción en múltiples líneas y, avise rápidamente de su detección.

Para su implementación ha sido necesario utilizar múltiples tecnologías e instaurar diferentes comunicaciones entre ellas. La detección de los errores se realizará a través de algoritmos de reconocimiento de imagen, que permitan inspeccionar el producto durante su paso por la cadena de producción y, localizar así los posibles defectos del producto.

Por último, el sistema implantado en las diferentes líneas de producción será configurable desde un único punto de acceso, desde el cual será posible conectarse y obtener información de su funcionamiento en tiempo real.

**Palabras clave:** Visión por computador, detección de objetos, estándares de calidad, OpenCV, microcontroladores.

---

---

## ABSTRACT

Nowadays the quality of the products is one of the most important key factors to obtain a differentiation from competitors and so achieve an important competitive advantage. New technologies allow us to reach a high-quality standard, however, little and medium companies can't use often these technologies because they are very expensive

In this project we can see the development of a tool with a low implantation cost and a huge scalability, that allow find and recognize possible product errors along production chain in multiples lines. Moreover, with that solution we can obtain a detection system that advice quickly of the errors found.

To implement that solution, we need to use a different type of technologies interconnected among them. Error detection is realised through image detection algorithms that allow inspect the product among production process and localize eventually errors.

In conclusion, the system installed in several production lines is configurable from one unique access point, from that access point the final user could connect to the lines and obtain production information in real time.

**Keywords:** Computer vision system, object detection, quality standards, OpenCV, microcontrollers.

---





# ÍNDICE GENERAL

---

índice general .....	VI
Índice de figuras.....	VIII
Índice de tablas .....	IX

---

Estructura del TFG.....	10
Glosario .....	11
1 Introducción .....	12
1.1 Motivación.....	13
1.2 Objetivos.....	13
2 Fundamentos teóricos.....	14
2.1 Visión por computador.....	14
2.2 Raspberry Pi.....	24
3 Análisis y requisitos del problema.....	28
3.1 Caso de estudio.....	29
3.2 Requisitos .....	30
3.2.1 Requisitos funcionales.....	30
3.2.2 Requisitos no funcionales.....	31
3.3 Solución propuesta .....	31
3.4 Presupuesto.....	34
3.4.1 Mando de control.....	35
3.4.2 Cámara inteligente .....	35
4 Diseño de la solución .....	36
4.1 Arquitectura del Sistema.....	36
4.2 Tecnologías empleadas.....	39
4.2.1 Hardware .....	39
4.2.2 Software .....	46
5 Desarrollo de la solución propuesta.....	50
5.1 Mando de control.....	50
5.1.1 Hardware .....	50
5.1.2 Software .....	55
5.2 Servidor .....	69
5.2.1 Hardware .....	70
5.2.2 Software .....	71



5.3	Software de apoyo.....	80
5.3.1	Configuración algoritmo detección de perímetro.....	81
5.3.2	Configuración algoritmo detección de producto .....	82
6	Pruebas.....	84
7	Conclusiones .....	86
7.1	Relación del trabajo desarrollado con los estudios cursados.....	88
	Agradecimientos .....	89
	Bibliografía .....	90
<hr/>		
	Anexos .....	92
	Diagrama GPIO Arduino Micro.....	92
	Configuración y prueba modulo cámara.....	93
	Imágenes montaje y funcionamiento del mando de control .....	95



## ÍNDICE DE FIGURAS

---

Figura 2-1. Esquema sistema de visión por computador industrial. ....	16
Figura 2-2. Funcionamiento cámara sensor lineal. Imagen extraída de [3]. ....	17
Figura 2-3. Efectos de la distancia focal larga. Imagen extraída de [7]. ....	18
Figura 2-4. Efectos de la distancia focal corta. Imagen extraída de [7]. ....	19
Figura 2-5. Distorsiones causadas por distancia focal corta. Imagen extraída de [3]. ....	19
Figura 2-6. Clasificación de unidades ópticas según sus parámetros. Imagen extraída de [7]. ....	20
Figura 2-7. Ejemplo de una correcta iluminación. Imagen extraída de [3]. ....	21
Figura 2-8. Iluminación a contraluz. Imagen extraída de [3]. ....	21
Figura 2-9. Iluminación de anillo. Imagen extraída de [3]. ....	22
Figura 2-10. Iluminación direccional. Imagen extraída de [3]. ....	23
Figura 2-11. Iluminación Darkfield, Imagen extraída de [3]. ....	23
Figura 2-12. Ejemplo efectos iluminación Darkfield. Imagen extraída de [3]. ....	24
Figura 2-13. Logo Raspberry Pi. Imagen extraída de [9]. ....	24
Figura 2-14. Raspberry PI 4. Imagen extraída de [9]. ....	26
Figura 3-1. Descripción etapas detección de movimiento. ....	32
Figura 3-2. Explicación detección del perímetro. ....	33
Figura 4-1. Arquitectura sistema cliente-servidor. ....	36
Figura 4-2. Diagrama de flujo lógica servidor. ....	37
Figura 4-3. Diagrama de flujo lógica cliente. ....	38
Figura 4-4. Raspberry Pi 3 modelo B+. Imagen extraída de [9]. ....	39
Figura 4-5. Logo y productos Arduino. Imagen extraída de [13]. ....	40
Figura 4-6. Arduino Micro. Imagen extraída de [13]. ....	41
Figura 4-7. Pi camera module v2. Imagen extraída de [14]. ....	43
Figura 4-8. Raspberry Pi Touch. Imagen extraída de [17]. ....	45
Figura 4-9. Entorno de desarrollo Processing con Logo. ....	46
Figura 4-10. Entorno de desarrollo Arduino IDE con logo. ....	47
Figura 5-1. Mockup del mando de control. ....	51
Figura 5-2. Elementos hardware que componen el mando de control. ....	51
Figura 5-3. Valores analógicos joystick en los ejes cartesianos. ....	52
Figura 5-4. Esquema resistencia pull up. Imagen extraída de [28]. ....	53
Figura 5-5. Esquema conexiones Arduino Micro. ....	53
Figura 5-6. Montaje componentes hardware en el mando de control. ....	54
Figura 5-7. Mando de control. ....	55
Figura 5-8. Datos enviados observados por el monitor del puerto serie. ....	56
Figura 5-9. Correlación entre movimiento del joystick y valor enviado. ....	57
Figura 5-10. Ventana de conexión de la aplicación. ....	58
Figura 5-11. Ventana de configuración del sistema. ....	60
Figura 5-12. Ventana de configuración del sistema en modo edición. ....	62
Figura 5-13. Diferentes aspectos del botón de selección de lado. ....	63
Figura 5-14. Explicación desplazamiento e incremento del lado de un área. ....	67
Figura 5-15. Servidor ensamblado. Imagen extraída de [9]. ....	70
Figura 5-16. Etapas del envío de una imagen desde el servidor al cliente. ....	72
Figura 5-17. Ejemplo funcionamiento del algoritmo de detección de movimiento. ...	74
Figura 5-18. Ejemplo funcionamiento algoritmo detección de perímetro. ....	76



Figura 5-19. Ejemplo funcionamiento del algoritmo de detección de producto. ....	79
Figura 5-20. Resultados aplicación del algoritmo de detección de perímetro sobre diferentes productos. ....	81
Figura 5-21. Ejemplo de resultados de ejecución software de apoyo para la detección de los productos. ....	83
Figura anexos-1. Diagrama GPIO Arduino Micro, parte nº1. Imagen extraída de [13]. ....	92
Figura anexos-2. Diagrama GPIO Arduino Micro, parte nº2. Imagen extraída de [13]. ....	93
Figura anexos-3. Montaje parte superior del mando de control. ....	95
Figura anexos-4. Montaje parte inferior mando de control. ....	96
Figura anexos-5. Mando de control operativo. ....	96

## ÍNDICE DE TABLAS

---

Tabla 1. Precios de ordenadores Raspberry Pi según modelo. ....	25
Tabla 2. Especificaciones técnicas Raspberry Pi 3 B+. ....	27
Tabla 3. Perfiles de stakeholders. ....	28
Tabla 4. Presupuesto mando de control. ....	35
Tabla 5. Presupuesto servidor-cámara inteligente. ....	35
Tabla 6. Especificaciones técnicas Arduino Micro. Tabla extraída de [13]. ....	42
Tabla 7. Especificaciones técnicas Pi camera module v2. ....	44
Tabla 8. Especificaciones técnicas Pi touch. ....	45
Tabla 9. Correlación acciones usuario en el mando de control – acciones a realizar en el Sistema. ....	66
Tabla 10. Errores comunes configuración cámara y sus respectivas soluciones. ....	95

---

---

## ESTRUCTURA DEL TFG

---

El presente trabajo está compuesto por 7 capítulos. Seguidamente se describen las temáticas que éstos van a abordar:

- Capítulo 1. Introducción: Presentación general de los argumentos que van a ser tratados en este proyecto. Asimismo, se explican las motivaciones que han impulsado su realización y los objetivos que se pretenden conseguir.
- Capítulo 2. Fundamentos teóricos: Se hace hincapié sobre el contexto tecnológico actual y se documentan las tecnologías más relevantes que son empleadas en el proyecto. En particular, se realiza una introducción general de las mismas, se describen sus componentes y se explica su funcionamiento y aplicación.
- Capítulo 3. Análisis y requisitos del problema: Se exponen las fases iniciales del proyecto; en ellas se explica el caso de estudio que se ha analizado y los resultados obtenidos bajo forma de requisitos. Consecuentemente, se ilustra la solución propuesta para resolver el problema detectado, destacando el funcionamiento y el presupuesto.
- Capítulo 4. Diseño de la solución: Se refiere al proceso en el cual se ha ideado el diseño acerca de cómo se ha de realizar la solución propuesta; en él se documentan en dos niveles de detalle las decisiones tomadas. Dichos niveles exponen la arquitectura sobre la cual se basa el sistema y, pone énfasis en las tecnologías empleadas tanto a nivel software como hardware.
- Capítulo 5. Desarrollo de la solución propuesta: Se describe el desarrollo de la solución propuesta y de los elementos que la componen, exponiendo el funcionamiento y la realización de las aplicaciones ideadas y de los componentes hardware utilizados.
- Capítulo 6. Pruebas: Se presentan las pruebas que se han realizado para verificar que la solución funciona correctamente y, los resultados obtenidos. Adicionalmente, se explican las problemáticas más relevantes observadas y las soluciones adoptadas.
- Capítulo 7. Conclusiones: En este último capítulo se realiza una síntesis sobre los objetivos alcanzados con la realización del proyecto y, se explica brevemente la relación del trabajo con los estudios realizados. Adicionalmente, se explica qué ha supuesto para el alumno, tanto a nivel personal como técnico, la realización de este trabajo.



## GLOSARIO

- API: Interfaz de Programación de Aplicaciones.
- ARM: Máquina Avanzada RISC.
- DSI: Interfaz Serie de Pantalla.
- GPIO: Pin entrada/salida de propósito general.
- GPU: Unidad de Procesamiento Gráfico.
- HSB: Matiz, Saturación y Brillo.
- IDE: Entorno de Desarrollo Integrado.
- IoT: Internet de las cosas.
- JSON: JavaScript Notación Objeto.
- NAS: Almacenamiento conectado a red.
- OpenGL: Librería Gráfica Abierta.
- PCB: Placa de Circuito Impreso.
- PWM: Modulación por Ancho o de Pulso.
- RGB: Rojo, Verde y Azul.
- SoC: Sistema en Chip.
- TCP: Protocolo de Control de Transmisión.
- UART: Receptor-Transmisor Universal Asíncrono.
- UDP: Protocolo de Datagrama de Usuario.



---

---

# CAPÍTULO 1

## INTRODUCCIÓN

---

Hoy en día, nos encontramos en un mundo siempre más interconectado y globalizado, donde las conexiones entre países con respecto a los intercambios de mercancías aumentan tanto en número como en rapidez. Donde antes, las limitaciones geográficas y espaciales, aseguraban a las empresas tener una cantidad de pedidos confirmados para sus productos y un número limitado de competidores, ahora ya no se dan estas condiciones.

Para asegurarse una ventaja competitiva sobre las demás empresas, uno de los factores claves, entre otros, es la calidad de nuestro producto.

Donde antes había personas comprobando que se respetasen determinados estándares de calidad, controlando que el producto no estuviese ni dañado ni defectuoso en algunos de sus componentes, o visionando que la línea de producción funcionase correctamente, hoy encontramos cámaras que emplean algoritmos de visión artificial para realizar estas tareas de manera más rápida y eficiente y, al mismo tiempo abaratando los costes de dichas tareas.

De esta manera no sólo nos aseguramos una alta calidad del producto, sino que también tenemos un sistema de control sobre la cadena productiva, para hacer frente a cualquier imprevisto que pudiera surgir, con el fin de mantener nuestra ventaja competitiva.

¿Todas las empresas pueden permitirse implantar estas nuevas tecnologías? Lamentablemente, los costes de implantación de algoritmos de visión artificial o de software de visión por computador, son demasiado elevados para gran parte de las empresas.

Para cubrir este segmento de empresas, tecnologías como Arduino<sup>1</sup>, Raspberry PI<sup>2</sup>, y OpenCv<sup>3</sup> permiten desarrollar sus propios sistemas de visión por computador a un coste asequible, aunque se tendrán que asumir los costes y el tiempo de desarrollo del software necesario para la cadena productiva. Por otro lado, hay que valorar positivamente la ventaja de realizar un sistema a medida para las necesidades reales de la propia empresa, lo que permite una mayor flexibilidad en su uso e implantación.

En definitiva, el mundo del reconocimiento de imágenes y su integración con la industria está en una fase de grandes avances; en los próximos años se espera la incorporación de dispositivos e instrumentos más asequibles a la vez que un crecimiento en potencia y fiabilidad, con el objetivo que implantar esta tecnología sea la norma y no una excepción.



## 1.1 MOTIVACIÓN

La realización de este trabajo ha sido dictada por los siguientes motivos:

- Concienciar a las personas implicadas, de que no es necesario invertir ingentes recursos para poder disponer de sistemas válidos aplicables a todo tipo de situaciones y aplicaciones, como por ejemplo la que se va a tratar: estándares de calidad en un proceso productivo.
- Aprender e investigar sobre los múltiples usos que puede tener la visión artificial y el reconocimiento de objetos hoy en día, y cómo su utilización puede cambiar las tareas tediosas y repetitivas que antes involucraban a parte del personal para su realización.
- Demostrar que los miniordenadores como Raspberry PI, a pesar de su bajo coste, no son simples juguetes, sino que pueden desarrollar numerosas funciones, entre otras las de poder ejecutar de manera continuada programas complejos en ambientes no aptos para un ordenador convencional, como en un entorno industrial.
- Sensibilizar sobre los usos de Arduino, no sólo como instrumentos de aprendizaje en relación con el funcionamiento de los microcontroladores, sino también como dispositivos preparados para un uso masivo en múltiples escenarios.

## 1.2 OBJETIVOS

Este proyecto, nace durante mi etapa de prácticas universitarias, en la empresa de alimentación Vicky Foods (Dulcesol), líder en las categorías de pastelería y bollería en España.

En una de sus numerosas cadenas productivas, hay constancia de un fallo reiterativo a causa del cual la producción no alcanza los niveles máximos. Cuando éste se verifica, un operario debe ir y solucionar el problema, pero no existe ningún sistema que le avise cuándo dicho fallo ocurre, ni con el cual poder observar la línea de producción sin estar ahí presencialmente.

Por lo tanto actualmente, un trabajador mientras desempeña sus tareas, adicionalmente también observa que la línea funcione correctamente. Ésta es una tarea repetitiva e innecesaria si se adopta un sistema de visión por computador, que permita identificar en tiempo real la verificación de fallos.

Dicho sistema puede detectar cuándo el paso de los productos se interrumpe o la producción disminuye, y posteriormente avisar a un operario para que éste pueda desde su puesto verificar la situación, sin tener que desplazarse hasta la línea y, actuar sólo si es realmente necesario.



Evaluando la situación, se ha decidido desarrollar una aplicación de esta tipología, que solucione el problema actual, ya que con su realización se conseguirían los siguientes objetivos:

- Automatizar la tarea repetitiva, descrita anteriormente, para aligerar la carga de trabajo de los operarios.
- Obtener un sistema escalable a las restantes líneas de producción manteniendo un bajo coste de instalación.
- Configurar cada software de reconocimiento, instalado en las diferentes líneas, desde un único punto de acceso.
- Observar en tiempo real el estado de la línea de producción desde remoto.

Adicionalmente, con la implantación de dicho sistema se verán reflejadas las siguientes ventajas.

- Servicio de aviso de errores en la línea productiva, operativo de forma continuada.
- Mayor eficiencia productiva.
- Incremento en la velocidad de subsanación de los errores detectados en la cadena de producción, debido a su detección inmediata.
- Reubicación de los operarios para trabajos especializados.
- Registro de los errores y de sus causas.
- Videovigilancia continuada de la línea.

---

## CAPÍTULO 2

### FUNDAMENTOS TEÓRICOS

---

En este apartado se mencionan algunas de las más importantes tecnologías empleadas en el proyecto, como por ejemplo el concepto de visión por computador, a la vez que se ilustran las ventajas sobre el uso de los ordenadores de placa reducida.

Se realiza una introducción al concepto de visión por computador, al igual que a las técnicas que se han estado empleando hasta la fecha y, que están siendo reemplazadas por las redes neuronales profundas.

#### 2.1 VISIÓN POR COMPUTADOR

La visión por computador, o visión artificial, es una de las ramas de la Inteligencia Artificial, concretamente es la disciplina compuesta por un conjunto de herramientas y métodos que permiten obtener, procesar y, analizar una sola imagen o una secuencia



de imágenes de forma automática, con la finalidad que puedan ser tratadas por un ordenador para conseguir y comprender información útil.<sup>[1]</sup>

“Desde la perspectiva de la ingeniería, busca automatizar las tareas que puede realizar el sistema visual humano”<sup>[2]</sup>. “Esto permite mecanizar una amplia gama de trabajos, al aportar a las máquinas la información que necesitan para la toma de decisiones correctas, en cada una de las labores para las cuales han sido designadas”.<sup>[1]</sup>

¿Qué función desempeña en los varios ámbitos en los que es empleada? Dentro de sus principales usos, destaco aquellos más involucrados con mi proyecto:

- **Detección de objetos:**  
Una vez los sistemas son entrenados y configurados a través de los patrones adecuados, éstos permiten la detección de objetos, ya que, cuando el ordenador recibe la información visual, puede relacionarla con los patrones aprendidos con anterioridad, y, de esta forma, identificar los objetos y diferenciarlos.<sup>[1]</sup>
- **Análisis de vídeo:**  
“Al mismo tiempo, al tener la capacidad de identificar cada uno de los objetos, esta habilidad se puede utilizar para llevar a cabo un análisis de vídeo; lo cual, es especialmente útil a la hora de aplicarlo en estrategias de seguridad y control.”<sup>[1]</sup>

Por consiguiente, estas técnicas tienen aplicaciones en todos los ámbitos industriales, como por ejemplo: automoción, alimentación, electrónica, medicina o logística, entre otros.

En concreto, en el sector de la alimentación, la visión artificial es un elemento fundamental en el control de calidad de los productos, especialmente en las fases finales. Tanto la detección de materiales ajenos no comestibles en los productos, como el correcto cierre de los envases, dependen directamente de sistemas de visión artificial que supervisan el desarrollo en la mayoría de sus fases.<sup>[1]</sup>

Independientemente del sector industrial donde se aplique, los sistemas de visión por computador en estos ámbitos, siguen un esquema estándar que se compone principalmente de los siguientes elementos:

1. Unidad de procesado de imágenes.
2. Cámara.
3. Unidad Óptica.
4. Iluminación.



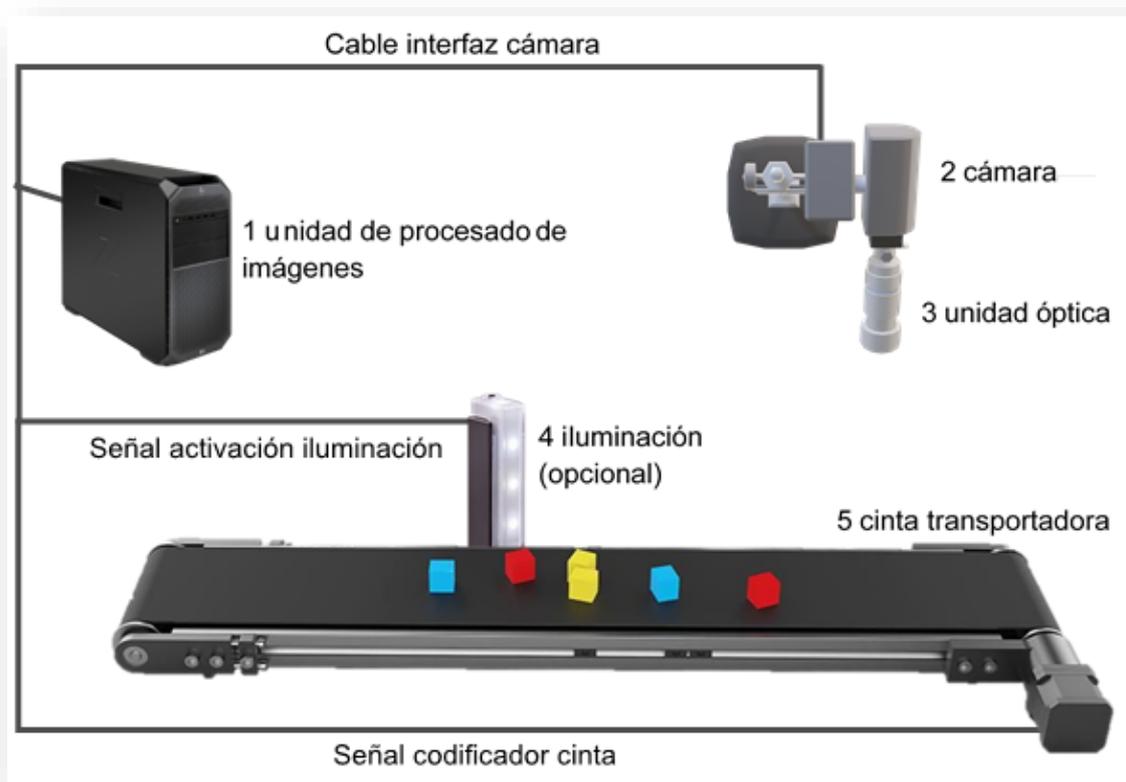


Figura 2-1. Esquema sistema de visión por computador industrial.

Veamos en detalle qué función realiza cada uno de los elementos dentro del sistema:

1. Unidad de procesamiento de imágenes.

Es el centro de mando del sistema; la unidad se encarga de procesar, analizar las imágenes que le son enviadas por la cámara y extraer la información necesaria de éstas, para luego redirigirla donde sea necesario.

Si el sistema no dispone de una cámara inteligente, que sepa reconocer en qué momento debe realizar una fotografía, será él mismo el encargado de enviar las señales que la activen.

Otra de sus funciones, si el sistema lo requiere, es la de enviar las señales de activación y paralización de la cinta que transporta los productos, con el fin que éstos queden en la posición óptima para ser fotografiados.<sup>[3]</sup>

Por último, si es necesario, gestiona el método de iluminación; este componente es opcional dentro del sistema, pero tiene una gran importancia y facilitará mucho la labor de procesamiento de la imagen. Más adelante volveré a hacer hincapié en ello.

## 2. Cámara.

“La función de las cámaras de visión es, capturar la imagen proyectada en el sensor para poder transferirla a un sistema electrónico y que pueda ser interpretada, almacenada y visualizada.”[\[4\]](#) En nuestro caso, el sistema es en la unidad de procesado de imágenes, un computador, pero bien puede tratarse de un monitor para visualizar la imagen.

Las cámaras de visión artificial se difieren de las cámaras normales gracias a que poseen unas características específicas, como el control del disparo para capturar las piezas que pasan por delante de la cámara, exactamente en la posición requerida. Además, ofrecen un control de los tiempos completo, así como un control de las señales y de la velocidad de obturación, de la sensibilidad y de otros factores fundamentales.[\[4\]](#)

Hoy en día debido a sus múltiples usos, podemos observar diferentes tipologías de cámaras de visión industriales, donde cada una de ellas desempeña una labor específica, como por ejemplo las cámaras infrarrojas o térmicas, las 3D o las de alta velocidad.

Otro método para diferenciar las diferentes tipologías de cámaras es, según el formato del sensor; basándonos en él obtenemos dos categorías:

- Sensor de área: Estas cámaras se caracterizan por tener un sensor más grande, de forma rectangular; su bajo coste y su facilidad de uso permiten su empleo a gran escala. Conjuntamente, aseguran tiempos de exposición más cortos.[\[3\]](#)
- Sensor lineal: Su sensor, como su nombre indica, es lineal y por lo tanto su funcionamiento es diferente de las demás cámaras. Mientras un objeto va avanzando y va entrando en el campo de visión de la cámara, ésta toma fotos lineares haciendo que la imagen sea una secuencia de líneas. Consecuentemente, tendremos una mayor resolución. Son utilizadas principalmente, para capturar con encuadres grandes; no obstante, su instalación es compleja.[\[3\]](#)



Figura 2-2. Funcionamiento cámara sensor lineal. Imagen extraída de [\[3\]](#).

En la actualidad, la mayoría de las cámaras empleadas en los sistemas de visión por computador no son las convencionales, sino las llamadas cámaras inteligentes.

Una cámara inteligente, incorpora los elementos y componentes tradicionales de las cámaras clásicas, como los sensores y la electrónica asociada a la captura de la imagen; conjuntamente engloban un procesador, memoria y sistema de comunicaciones con el exterior a través de diferentes canales. Es decir, se consideran como un sistema completo de visión artificial al integrar todos los elementos propios de éste en tamaño reducido.[5]

### 3. Unidad óptica.

La unidad óptica está formada por una lente o un grupo de lentes, que proyectan la luz del exterior en el sensor de la cámara y permiten establecer el ángulo de apertura de la escena. Dicho ángulo es la magnitud que determina la parte de la escena que es captada en el sensor.[3]

Generalmente, el sistema visual humano nos permite captar mucha más escena de la que se ve reflejada en las fotos. No obstante, consecuentemente al tipo de lente utilizada y a su distancia focal, podemos observar cómo varía el tamaño de esta porción.[6]

La distancia focal es medida en milímetros, y es la distancia entre el centro óptico de la lente y el sensor de la cámara; variando esta última lograremos los siguientes efectos sobre la fotografía:[3]

A mayor distancia focal obtendremos una menor área de encuadre para una misma distancia de la cámara a la escena; de la misma manera el ángulo de visión se reducirá . Adicionalmente conseguiremos menor profundidad de campo, de forma que aislaremos al sujeto del fondo.

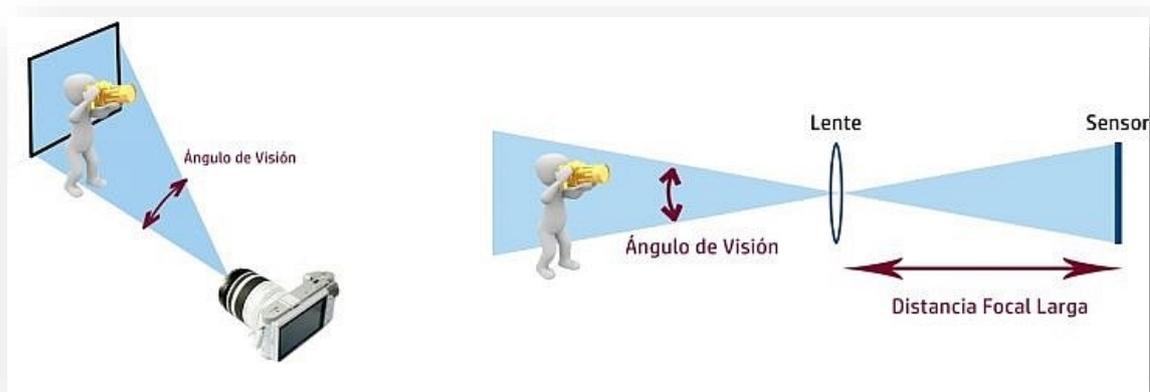


Figura 2-3. Efectos de la distancia focal larga. Imagen extraída de [7].

A menor distancia focal obtendremos una mayor área de encuadre para una misma distancia de la cámara a la escena. Asimismo, aumentará el ángulo de visión y la profundidad de campo. Esto significa que tendremos más elementos de la fotografía a foco.

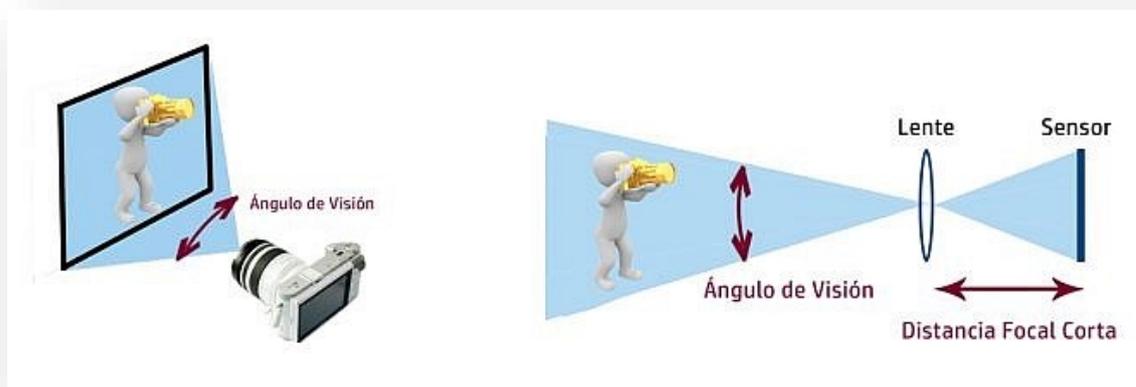


Figura 2-4. Efectos de la distancia focal corta. Imagen extraída de [7].

Una distancia focal muy corta puede producir distorsiones en la lente; dichas distorsiones toman el nombre de acerico o de barril según el efecto indeseado que realicen.[3]

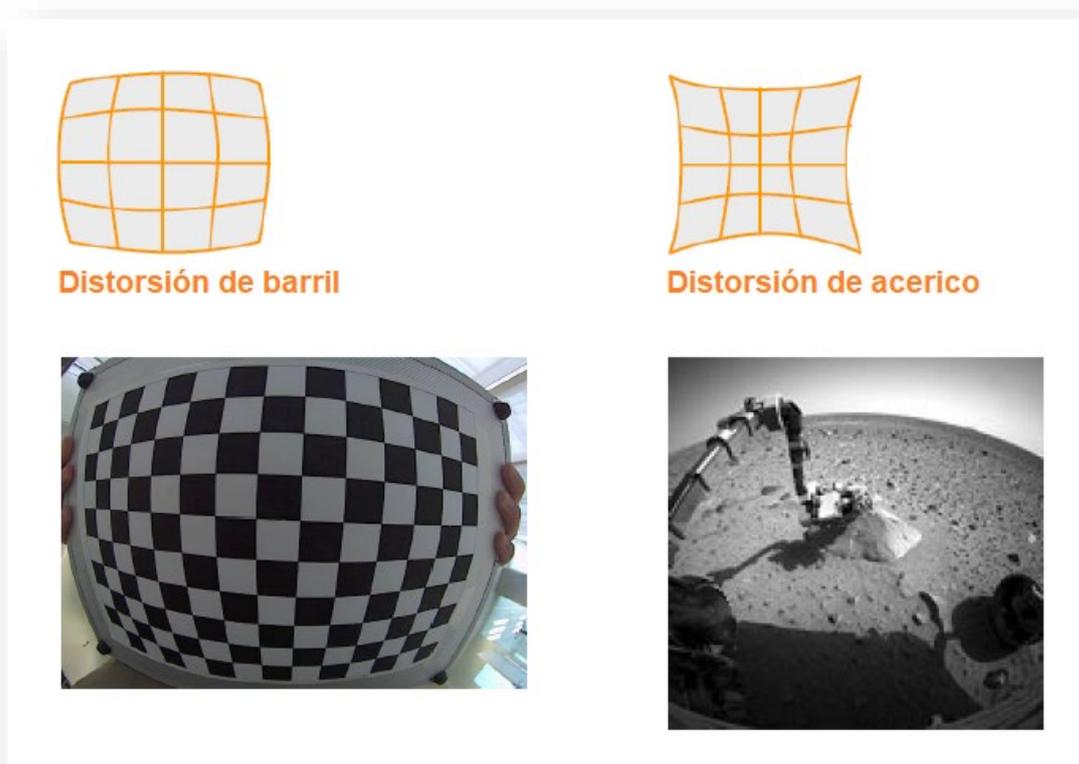


Figura 2-5. Distorsiones causadas por distancia focal corta. Imagen extraída de [3]

Para eliminar el error producido por el ángulo de visión y magnificación (la relación entre el tamaño de la imagen y el tamaño del objeto), en las aplicaciones que se especializan en la medición de precisión, recurriremos a ópticas telecéntricas.[3]

Según sus características, las unidades ópticas toman nombres diferentes y sirven para funciones distintas. Por lo tanto, es muy importante elegir adecuadamente la óptica más apta en función del entorno de trabajo en el que nos encontramos, y lo que deseemos realizar con ella.

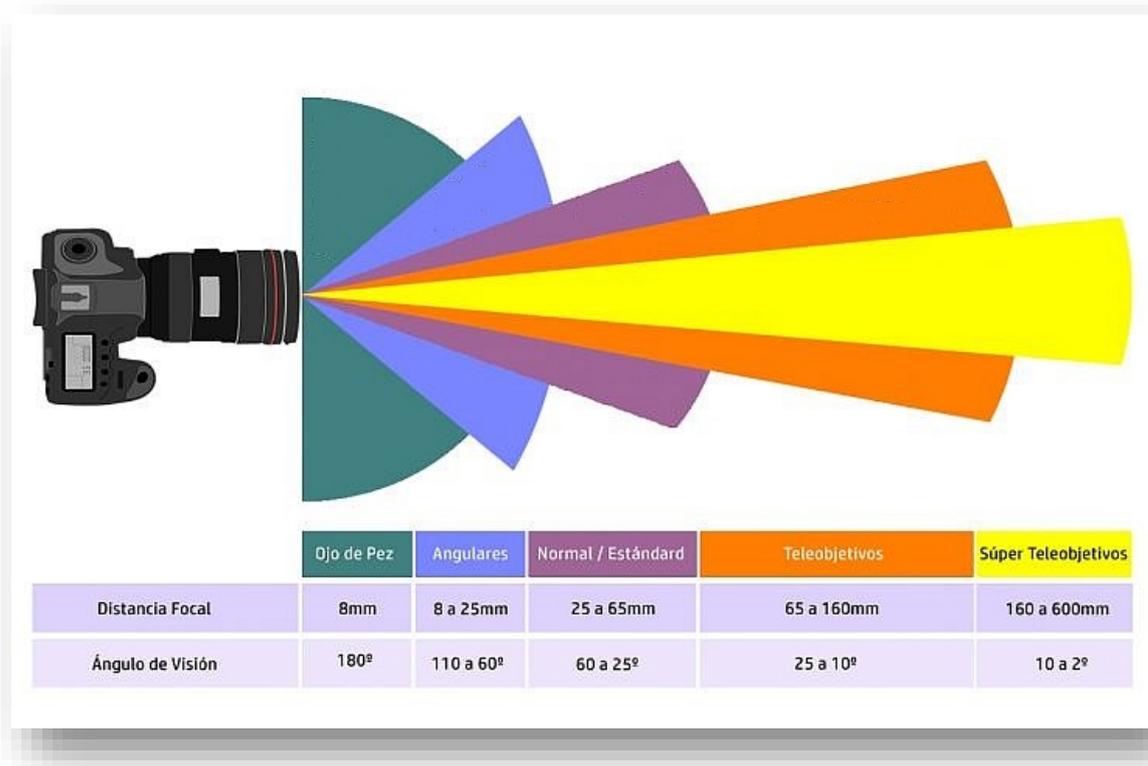


Figura 2-6. Clasificación de unidades ópticas según sus parámetros. Imagen extraída de [7]

#### 4. Iluminación.

En la base de un buen sistema de visión por computador, se encuentra la capacidad de distinguir el objeto deseado entre otros; ello se traduce en la habilidad para distinguir los píxeles de interés, (así definidos porque forman parte de la imagen que representa lo que queremos reconocer), de los que no son de interés.[3]

A pesar de ser opcional, y no estar presente en todos los sistemas, dicho componente es de importancia vital, pues favorece y simplifica el trabajo de procesado y su posterior análisis, entregando una imagen en las mejores condiciones posibles ya que aporta una buena iluminación, ergo:

- Mantiene constante la dirección y la intensidad de la luz.
- Optimiza el contraste.
- Separa el objeto de interés del fondo de la imagen.

A continuación, podemos ver un claro ejemplo de cómo una correcta iluminación agiliza y reduce dicho trabajo de procesado. La fecha de caducidad mostrada en la primera imagen, es difícil de reconocer debido a los reflejos provocados por la luz y el material del objeto; estos reflejos, pueden anularse con una iluminación adecuada, con el resultado que obtendremos una imagen bien definida y sencilla de analizar.

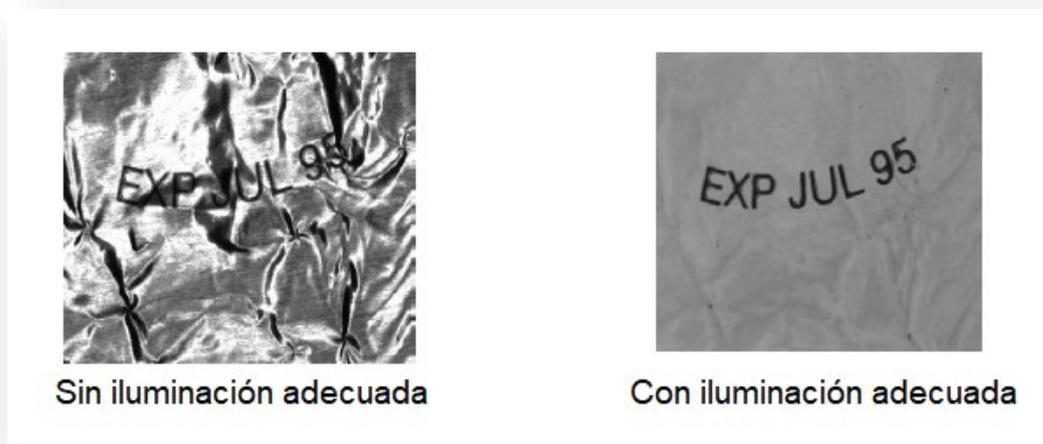


Figura 2-7. Ejemplo de una correcta iluminación. Imagen extraída de [3].

Existen diferentes técnicas de iluminación; la elección de un método respecto a otro dependerá de las características del objeto a iluminar. Dichos procedimientos se agrupan en dos categorías:

A contraluz.

En esta técnica, la fuente de iluminación se encuentra opuesta a la cámara, mientras que el objeto de interés se encuentra ubicado entre la luz y la cámara. Es empleada principalmente, en aplicaciones de localización y medición de objetos, ya que permite resaltar los contornos y, con objetos transparentes ayuda a destacar su estructura interna.[3]



Figura 2-8. Iluminación a contraluz. Imagen extraída de [3].

Directa.

Esta metodología se caracteriza a causa de la iluminación, que posee la misma orientación que la cámara y, a su vez se divide en diferentes tipologías. Brevemente cito las más relevantes:

- Anillo: Se utiliza para resaltar las características de la superficie en partes no reflectantes y, conjuntamente reducir las sombras. Ello es posible debido a que, se realiza un centrado de la luz en la imagen, gracias a que dicha luz proviene de un dispositivo con forma circular.[3]

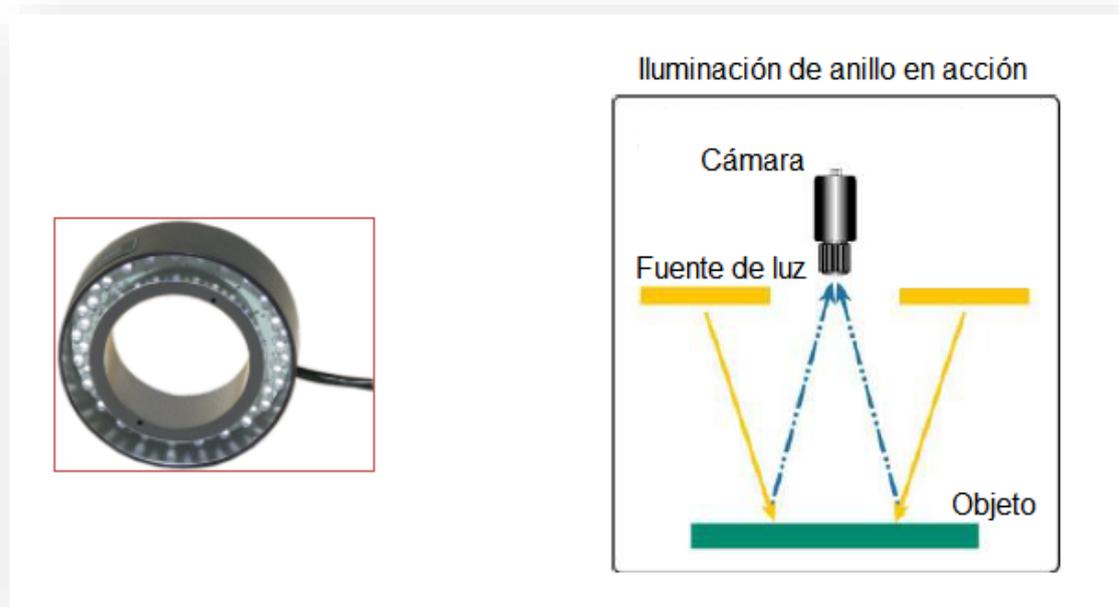
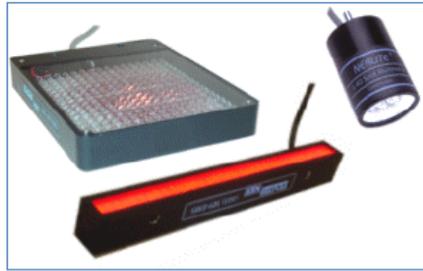


Figura 2-9. Iluminación de anillo. Imagen extraída de [3].

- Direccional: Utilizada principalmente en aplicaciones no críticas, puesto que la luz es unidireccional, posee un bajo coste de implantación. Debido a la característica de que la luz viene de una única dirección, esta metodología es empleada para la iluminación de superficies sin brillos, ya que de lo contrario, en superficies brillantes, ésta se vería reflejada y sería ineficaz.[3]



Emisores de luz direccional

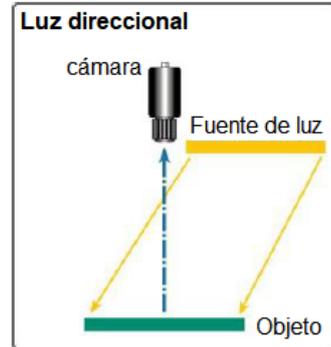


Figura 2-10. Iluminación direccional. Imagen extraída de [3].

- Darkfield: Dicha técnica debe su nombre al hecho que, la luz en segundo plano no es captada por la cámara, con el resultado que obtendremos un fondo obscuro o negro; así, empleará una iluminación angular para resaltar irregularidades en diferentes superficies; en virtud de la luz que impacta el objeto en manera oblicua, desde varias direcciones, podremos enfatizar las características en el plano de incidencia, como observaremos a continuación.[3]

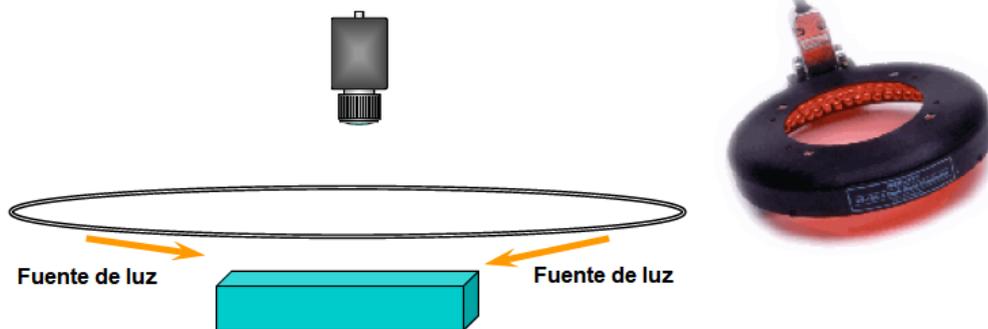


Figura 2-11. Iluminación Darkfield, Imagen extraída de [3].

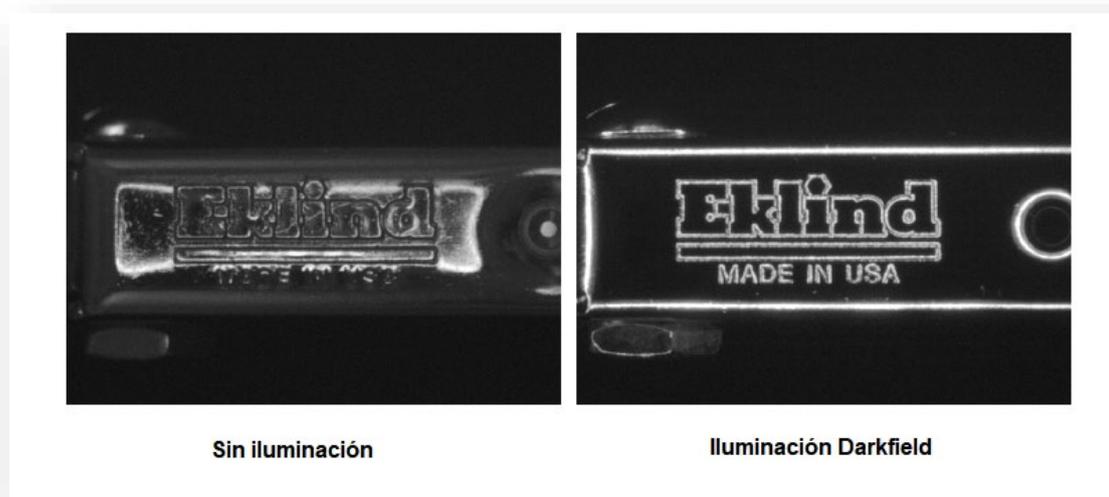


Figura 2-12. Ejemplo efectos iluminación Darkfield. Imagen extraída de [3].

## 2.2 RASPBERRY PI

Desde la aparición de los computadores en nuestras vidas hasta hace una década, adquirir un ordenador suponía un gasto considerable. No obstante, con el pasar de los años y a través del avance de la tecnología y la fabricación en masa, se han ido abaratando sensiblemente los costes.

En 2006 se empezó a diseñar y, posteriormente en 2012 se lanzó a la venta al público un producto que en poco tiempo ya es conocido y utilizado globalmente, además de ser en la actualidad la tercera marca de ordenadores más vendida en el mundo: Raspberry Pi.

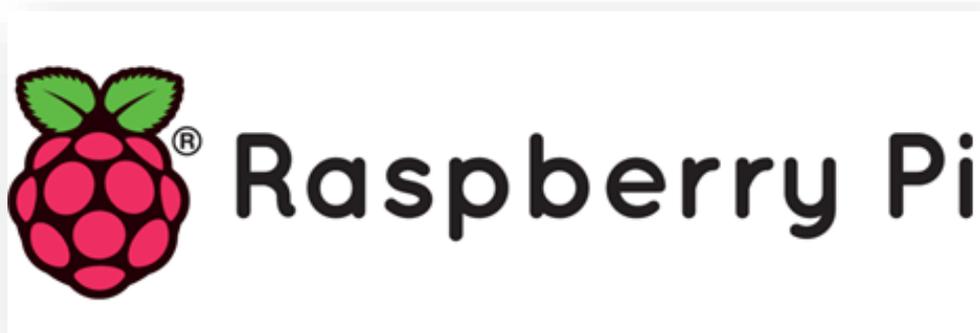


Figura 2-13. Logo Raspberry Pi. Imagen extraída de [9].

Raspberry Pi es un ordenador de placa reducida o única, de las dimensiones de una tarjeta de crédito, desarrollado en el Reino Unido por Raspberry Pi Foundation.[\[8\]](#)

El objetivo de dicha Fundación es, impulsar la educación de adultos y niños en el ámbito informático y del desarrollo del software, incluyendo los campos interrelacionados para que en un futuro puedan disponer de las mejores oportunidades a nivel laboral.[\[9\]](#)

Para llegar a este objetivo, proporciona globalmente acceso al mundo de la computación sirviéndose de ordenadores de bajo coste unidos a software libres y de altas prestaciones. La versión base de gama baja, a fecha de la redacción de este documento, cuesta solamente 5 \$ mientras que la versión que ofrece más prestaciones tiene un precio de 55 \$.

Familia	Modelo	Fecha lanzamiento	Precio
Raspberry Pi 1	A+	11/2014	20\$
	B+	07/2014	25\$
Raspberry Pi 2	B	02/2015	35\$
Raspberry Pi 3	A+	11/2018	25\$
	B	02/2016	35\$
	B+	03/2018	35\$
Raspberry Pi Zero	Zero	11/2015	5\$
	W	02/2017	10\$
Raspberry Pi 4	B	06/2019	35\$-55\$ *

\* Dependiendo de la configuración deseada.

Tabla 1. Precios de ordenadores Raspberry Pi según modelo.

A pesar de los bajos precios, los ordenadores de Raspberry pi mantienen todas las funcionalidades de los computadores convencionales; es suficiente conectar una pantalla y un teclado para interactuar con ella, exactamente igual que con cualquier otro PC.

Adicionalmente, Raspberry incorpora funciones de electrónica como pines GPIO, y de comunicación como UART. Estas funciones hacen que pueda ser empleado en proyectos de electrónica y robótica interactuando con sensores (temperatura, luz, aceleración...) y actuadores (motores, servos, relés...). [\[10\]](#)

No obstante, hay que remarcar que la potencia de estos minicomputadores no es equiparable a la que disponemos en un ordenador convencional; de hecho, observando las prestaciones del modelo más prestante de Raspberry, podemos corroborar que éste es equiparable a un ordenador de gama media del año 2007, o si decidimos compararlo con un ordenador actual de la misma gama, es 6 veces más lento.[\[11\]](#)

Sin embargo, quiero hacer hincapié sobre el hecho que sólo el procesador de dicho computador hodierno tomado como referencia, cuesta tres veces más que el ordenador de Raspberry.



Por añadidura, otro punto fuerte de Raspberry es su comunidad de usuarios, ya que proporciona una gran cantidad de desarrollos, documentación y tutoriales, que colaboran a la introducción y popularización del dispositivo.

Esta tipología de miniordenadores ejecuta generalmente sistemas operativos libres basados en Linux, aunque también es posible utilizar Windows 10 IoT, una versión libre del conocido Windows especializada para sistemas embebidos.

¿Exactamente qué podemos realizar con un Raspberry PI?

Se podría pensar que su uso no se aleja del objetivo para el cual fue concebido, el del aprendizaje. No obstante, existen infinidad de usos y proyectos que podemos llevar a cabo con un Raspberry.

Además de como primera toma de contacto con el mundo de la programación, éstos son usados principalmente como centros multimediales, servidores NAS, ordenadores Linux y por último videoconsolas.

Las aplicaciones menos habituales, pero al mismo tiempo más interesantes, incluyen todo tipo de proyectos de electrónica y robótica, aplicaciones de visión por ordenador, aplicaciones en IoT y domótica. Adicionalmente es posible conectar varios Raspberry Pi en paralelo para hacer un pequeño clúster de computación.[\[10\]](#)

Se podría deducir que todas estas aplicaciones serían aptas para un uso personal más bien que industrial. Sin embargo, gracias a los avances de los últimos años en esta tecnología se ha podido llegar a la conclusión que Raspberry Pi ya no se limita exclusivamente a proyectos científicos o personales, sino que puede ser usada ampliamente como una solución industrial IoT para controlar zonas de producción, automatizar procesos, control de calidad... y, alcanzar los objetivos de la industria 4.0.

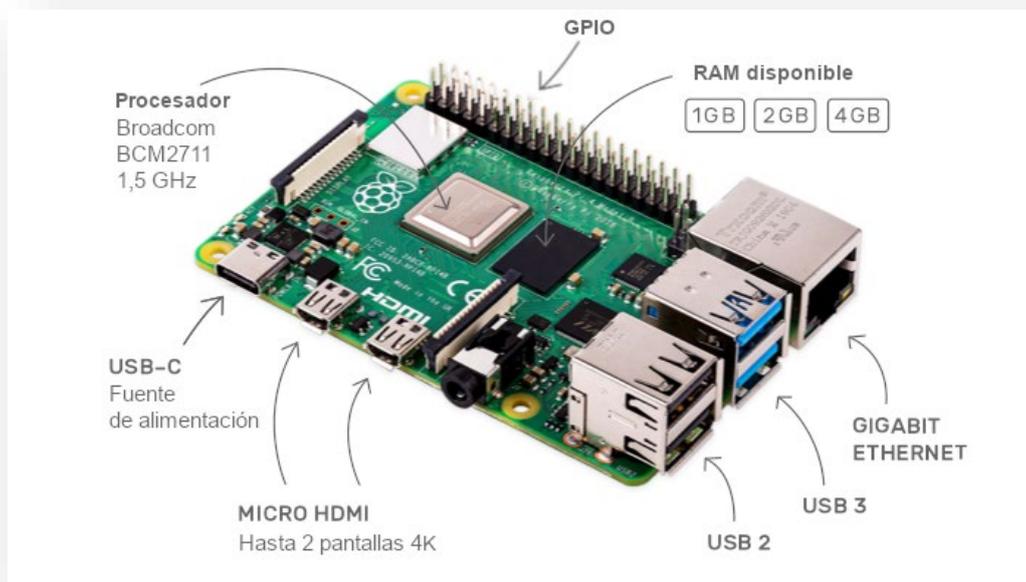


Figura 2-14. Raspberry PI 4. Imagen extraída de [\[9\]](#).

A la fecha inicial de este proyecto, el último modelo, el Raspberry Pi 4 mostrado en la imagen anterior, resultaba agotado debido al gran número de pedidos, por lo que se va a utilizar el modelo anterior hasta que sea posible sustituirlo con el nuevo, ya que este último implementa notables avances técnicos.

Nuestro referente, de la familia Raspberry pi 3 , toma la nomenclatura B+ y posee las siguientes especificaciones técnicas:

<b>Especificaciones técnicas Raspberry Pi 3 B+</b>	
<b>Procesador</b>	Procesador de núcleo cuádruple Broadcom BCM2837 A53 (ARM v8) SoC de 64 bits.
<b>Memoria</b>	SDRAM LPDDR2 de 1 GB.
<b>Bluetooth</b>	2,4 Ghz / 5,0 GHz IEEE 802.11ac.
<b>Ethernet</b>	Gigabit Ethernet a través de USB 2.0 (300 Mbps máx.).
<b>USB</b>	Cuatro puertos USB 2.0.
<b>Conexión</b>	Conector GPIO 40 contactos.
<b>HDMI</b>	Uno a tamaño completo.
<b>Vídeo</b>	DisplayPort MIPI DSI, puerto para cámara MIPI CSI, puerto de vídeo compuesto y salida estéreo de 4 polos.
<b>Multimedia</b>	Decodificador H.264, MPEG-4 (1080p30). Codificación H.264 (1080p30). Gráficos OpenGL ES 1.1, 2.0.
<b>Almacenamiento</b>	Ranura de tarjeta microSD para cargar sistema operativo y almacenamiento de datos.
<b>Alimentación</b>	Conector USB para 5,1 V / 2,5 A, corriente continua.

*Tabla 2. Especificaciones técnicas Raspberry Pi 3 B+.*

---

---

## Capítulo 3

### ANÁLISIS Y REQUISITOS DEL PROBLEMA

---

En este capítulo se van a describir las fases iniciales del proyecto, dichas fases comprenden la elicitación y especificación de los requisitos entre otras, concluyendo con la solución propuesta para abordar el problema.

La elicitación es la primera actividad en el desarrollo del software y consiste en determinar de dónde vienen los requisitos y cómo obtenerlos. Para ello es necesario identificar stakeholders relevantes, los documentos informativos accesibles provenientes del sistema actual manual o automatizado y, por último fuentes de información externa.

La correcta identificación del propósito del sistema para satisfacer las necesidades de sus usuarios es fundamental, ya que si no, obtendríamos un producto erróneo que no sería de utilidad al cliente.

Para identificar las partes interesadas de las cuales se extraerán los requisitos, se han de considerar todas las personas relacionadas con el proyecto; el resultado nos llevará a obtener los siguientes perfiles:

Nombre y/o rol	Usuario directo	Intereses en el proyecto
Trabajador de la línea	✓	Mejorar y agilizar las tareas de control de calidad del producto y de intervención, en caso de malfuncionamiento.
Supervisor de la línea	✓	Obtener rápidamente información detallada, sobre fallos en la línea y su estado general.
Supervisor de la planta	✓	Tener bajo control el estado de todas las líneas y, poder visionar en caso de necesidad una línea en concreto.
Responsable producción	✗	Reducir el tiempo de intervención en caso de malfuncionamiento, para obtener así un aumento de la producción.

*Tabla 3 Perfiles de stakeholders.*

La documentación propia de la empresa que se ha analizado para obtener información Incluye: documentos accesibles provenientes del sistema actual manual y automatizado, guías del usuario, manual de operaciones y material de entrenamiento.

### 3.1 CASO DE ESTUDIO

Actualmente, en una cadena de producción del producto X, la masa viene vertida en cavidades presentes en bandejas rectangulares de tamaño estándar. Dichas bandejas vienen transportadas dentro de un horno industrial donde la masa cuece.

Las bandejas, a pesar de tener un formato estándar no siempre tienen la misma orientación en la cinta, sino que pueden ser transportadas con diferentes angulaciones; no obstante, estos valores se encuentran restringidos por unos umbrales máximos y mínimos .

Después de la fase de horneado, las bandejas son transportadas hasta un punto de control donde un mecanismo las detiene y las libera sistemáticamente, después de un cierto intervalo de tiempo, para que pasen a las siguientes fases.

Durante el recorrido hasta dicho punto pueden ocasionarse malfuncionamientos en la cadena transportadora que no permitan el avance de las bandejas; la empresa requiere un sistema automatizado que, cuando se verifique esta situación, la detecte automáticamente y envíe un aviso de malfuncionamiento del proceso.

Adicionalmente, el sistema de vertido puede presentar irregularidades; como consecuencia de ello puede verificarse que no todas las cavidades de las bandejas sean rellenas, o que circulen por la línea bandejas vacías.

Por lo tanto, se requiere que el sistema propuesto como solución al problema pueda detectar el número exacto de productos contenidos en una bandeja, y si dicho número está por debajo de un umbral dado, se considere como error y se realice el aviso de malfuncionamiento del proceso.

El aviso de malfuncionamiento debe ir acompañado de una foto ilustrativa del problema detectado, para que sea de fácil comprensión.

El sistema que se requiere debe poder ser escalable a todas las líneas de producción; adicionalmente, su configuración debe efectuarse de forma ágil por los trabajadores desde remoto, sin tener que acceder in situ al lugar donde está instalado el dispositivo.

Se debe tener en cuenta que, en una misma línea, se pueden fabricar diferentes productos con variaciones en tiempos de producción, número de productos por bandeja, tamaño, y forma. Además, que la cinta transportadora puede funcionar en ambas direcciones.

Debido a que en estos momentos se está realizando un estudio, que indicará la mejor manera de integrar esta nueva aplicación con el sistema de informes y malfuncionamientos existente, las imágenes y avisos se guardarán temporalmente en la memoria local de la nueva aplicación.



## 3.2 REQUISITOS

Tras un atento análisis de los documentos a disposición, (extrapolación de las informaciones obtenidas a través de entrevistas con las partes interesadas citadas anteriormente), se han recogido los siguientes requisitos que la aplicación debe satisfacer y que se dividen en:

- Requisitos funcionales, que describen la funcionalidad o los servicios que se espera que el sistema proveerá conjuntamente a sus entradas y salidas.
- Requisitos no funcionales, que hacen referencia a las propiedades emergentes (atributos) del sistema, como la fiabilidad, tiempo de respuesta, capacidad de almacenamiento, capacidad de los dispositivos de entrada/salida, ajuste a estándares, etc.

### 3.2.1 Requisitos funcionales

- El usuario podrá conectarse a cualquier dispositivo puesto sobre la línea de producción.
- El usuario podrá configurar un dispositivo en una línea dada.
- El usuario debe poder ver en tiempo real lo que está ocurriendo en la línea de producción.
- Si lo desea, para facilitar la configuración, el usuario debe poder pausar y reanudar la conexión vídeo en tiempo real con el dispositivo.
- Durante la configuración se debe poder indicar el sentido de marcha de la cinta transportadora.
- El sistema tendrá dos modalidades: Configuración y Operativa; en la primera, esperará a recibir nuevas configuraciones y mostrará en tiempo real las imágenes que está captando; en la segunda, el sistema analizará todos los productos que estén pasando.
- El sistema deberá ser capaz de reconocer las bandejas transportadas independientemente de su orientación.
- El sistema deberá ser capaz de contar el número de productos válidos en cada bandeja.
- El sistema, si detecta menos productos que el umbral dado, deberá considerar lo que está ocurriendo como una anomalía.
- El sistema, en caso de detección de una anomalía, deberá almacenar la fotografía descriptiva del problema.



### 3.2.2 Requisitos no funcionales

- La memoria del servidor deberá tener capacidad suficiente para poder almacenar correctamente tres avisos de error al día, durante un mes.
- Para conectarse a un dispositivo se deberá proporcionar ID y puerto de este último.
- El dispositivo para poderse conectar a las aplicaciones en las líneas deberá ser inalámbrico y portátil.
- El sistema debe poder ser escalable a las restantes cadenas de producción.
- La aplicación propuesta debe tener un coste contenido, respecto a los sistemas actuales fabricados por terceros.

## 3.3 SOLUCIÓN PROPUESTA

Para solucionar el problema planteado se propone la creación de la siguiente aplicación compuesta por una cámara inteligente, que adicionalmente desempeñará el rol de servidor, y por un mando de control.

El mando de control integrará una pantalla táctil, y gracias a él se podrá acceder a las diversas cámaras instaladas en las diferentes cadenas de producción, con el fin de ver en tiempo real el comportamiento de éstas y poder configurarlas.

La cámara inteligente estará constantemente analizando en tiempo real lo que está grabando, para abordar el problema de la detección de las bandejas; el software de dicha cámara se valdrá de tres áreas de detección de movimiento configurables desde el mando de control. Dichas superficies serán puestas estratégicamente al principio, centro y, fin del paso de las bandejas.

Para conseguir activar el proceso de captura de la imagen durante el paso de una bandeja, en cada una de dichas áreas se deberá verificar la condición de que se está registrando, en el sentido de marcha de la cinta, el movimiento de más de la mitad de los píxeles que las componen.

Esto quiere decir que para pasar a la siguiente etapa del proceso evitando falsos positivos, es necesario que se cumplan los siguientes pasos:



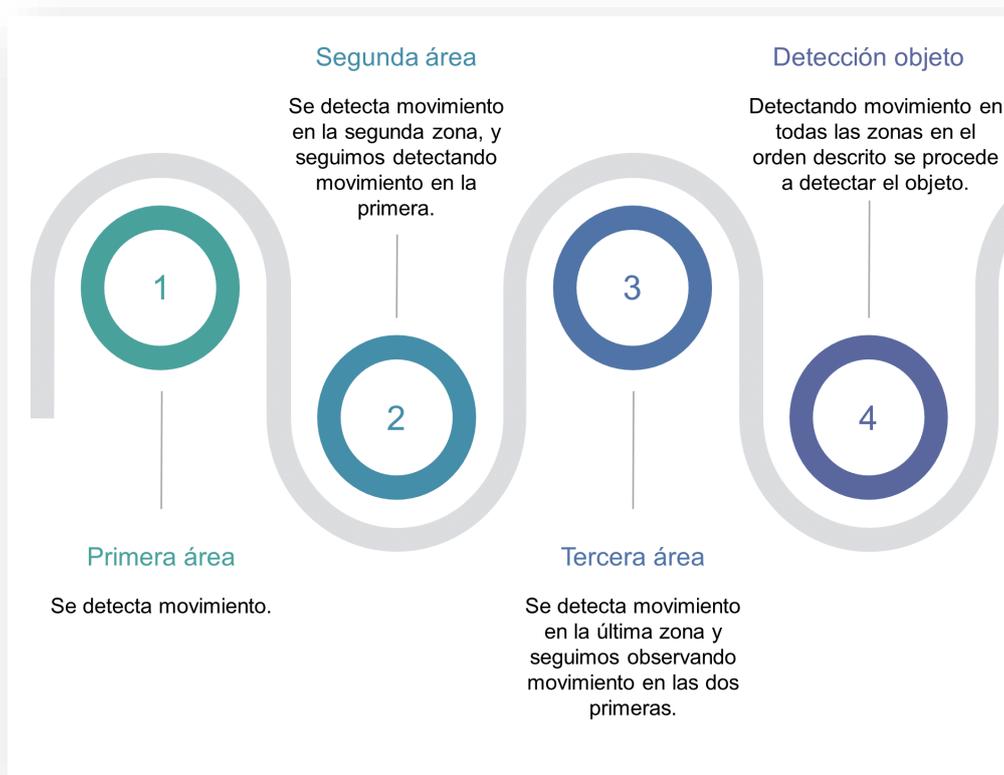


Figura 3-1. Descripción etapas detección de movimiento.

La detección de movimiento simultáneo en las tres áreas, activadas según el orden de marcha de la cinta transportadora, significa que está pasando un objeto en ese preciso instante y por lo tanto se procede a la detección de dicho objeto.

La siguiente etapa, implica detectar que el objeto que se está moviendo sea efectivamente la bandeja rectangular esperada. Conocemos las dimensiones de ésta, y por lo tanto podemos reconocerla encontrando un objeto que tenga las mismas dimensiones.

Para ello, se procede a buscar el perímetro del objeto; en el caso de que efectivamente sea la bandeja en cuestión, deberíamos encontrar 4 lados, donde cada lado es una línea recta de dimensiones conocidas.

El sistema busca estas 4 líneas rectas, no considerando las que tengan longitud diferente de la esperada; para saber en qué sector del vídeo debe focalizar la búsqueda, se le indican dos áreas rectangulares.

Las áreas se encuentran encapsuladas una dentro de la otra. El área más externa tiene una superficie ligeramente más grande que el tamaño de las bandejas, mientras que la interna tiene una dimensión inferior.

El espacio que se crea entre ellas delimita la superficie donde se prevé que vayan a encontrarse los bordes de las bandejas y por lo tanto, las líneas que forman su perímetro.

La aplicación buscará aquí dichas líneas; considerando que las bandejas pueden llegar con orientación diferente, se admitirán líneas con diferentes angulaciones, siempre que éstas empiecen y acaben dentro de la superficie entre las dos áreas, y su ángulo no sea, superior o inferior, a un umbral dado.

A continuación, podemos observar una ilustración sobre el mecanismo y, como se puede apreciar con todo detalle, los perímetros de las bandejas con diferente angulación se encuentran en la superficie de detección, y ninguno de ellos sobrepasa ni el área interna ni la externa.

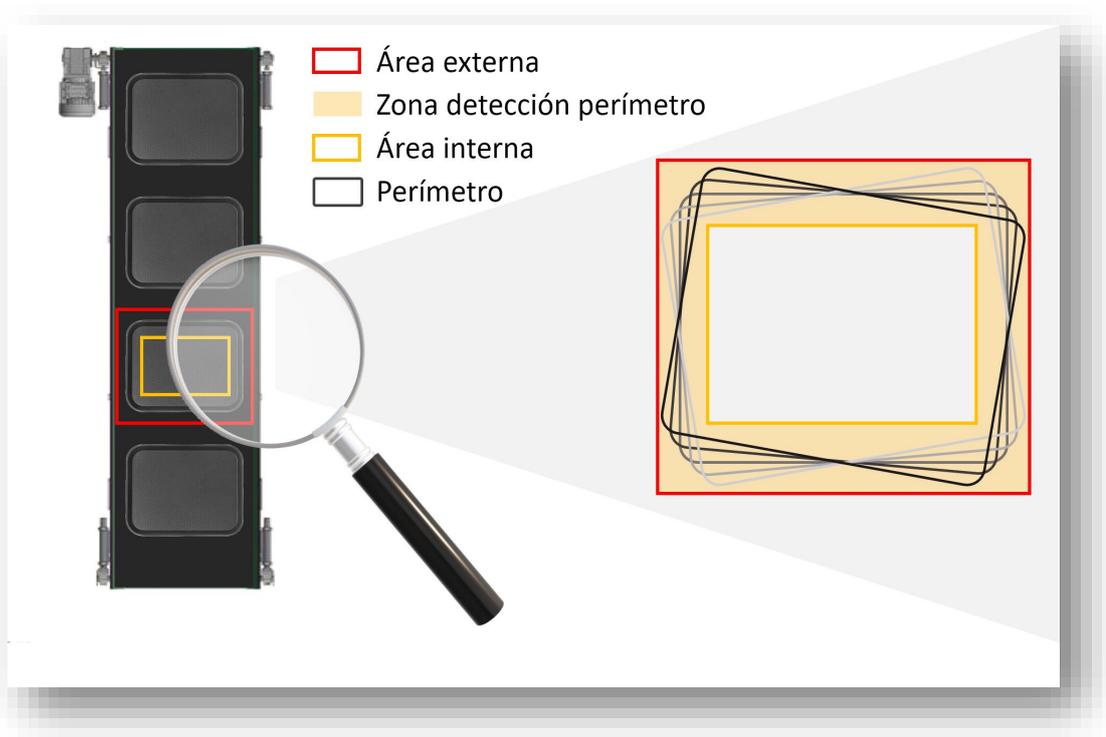


Figura 3-2. Explicación detección del perímetro.

Si no se encuentra el perímetro de la bandeja, se considera la detección de movimiento como un falso positivo. En caso contrario, si se logra su identificación se procede a analizar la imagen.

La primera operación del análisis consiste en separar los píxeles de interés de los de no interés; como he descrito anteriormente en el capítulo de “Fundamentos teóricos”, sección “Visión por computador”, ésta es una operación muy importante.

Para ello se va a utilizar un método de diferenciación basado en la detección de colores (más adelante hablaré con detalle sobre ello). En general, se separan los píxeles de los productos de lo que es su contenedor, la bandeja en este caso, a través del color.

Después, los píxeles con las mismas tonalidades son agrupados por proximidad, y es calculada su dimensión y forma. Si la forma y las dimensiones coinciden con las del producto que tiene que ser transportado, éste se suma al número de productos encontrados.

Si el número de productos encontrados es menor a un umbral dado, se procede a capturar la imagen y guardar esta última en la memoria interna del servidor.

### **3.4 PRESUPUESTO**

Uno de los requisitos establecidos para la implementación de esta solución, es su coste reducido en comparación con sistemas existentes de terceros; la solución propuesta respeta este criterio, no obstante, puede haber diferencias de prestaciones con aplicaciones más dispendiosas.

Consultando los diferentes proveedores del sector, se ha concluido que los precios de adquisición de una cámara industrial necesaria para el proyecto, tienen un valor que oscila entre 350 - 600 €. Adicionalmente, se debe incluir el coste del sistema de control.

El presupuesto del proyecto se divide según sus elementos; uno, destinado al mando de control y otro, destinado a la cámara inteligente, ya que sólo es necesario un único mando de control, mientras que las cámaras pueden variar en número.

Los importes indicados, se han obtenido seleccionando proveedores que ofrecían menor tiempo de entrega del producto, por lo tanto, considerando proveedores diferentes se pueden obtener diferencias a la baja significativas. Además, dichos importes son proporcionados de manera indicativa, ya que puede haber variaciones en sus valores.

El componente más costoso, es el mando de control, en proporción 2 a 1 respecto a la cámara; ello presenta una serie de ventajas, ya que sólo es necesaria una unidad para supervisar dichas cámaras; consecuentemente, el coste de adaptación a todas las líneas no es elevadamente oneroso, ya que la cámara tiene un coste de 69,90 €.

Los importes reflejados, y las unidades de los componentes, son las necesarias para construir respectivamente una unidad del mando de control y una unidad de la cámara inteligente.

En conclusión, el coste total del proyecto, incluyendo un mando de control y una cámara inteligente es de 225,69 €. No se ha incluido el importe de fabricación de la carcasa del mando de control, realizada a través de impresión 3D, al tener disponible esta tecnología en la empresa.



### 3.4.1 Mando de control

Producto	Precio unitario	Unidades	Precio total
Raspberry Pi 3 B+	35,65 €	1 Ud.	35,65 €
Pantalla táctil de 7 pulgadas	57,09 €	1 Ud.	57,09 €
Joystick	3,95 €	1 Ud.	3,95 €
Pulsador	4,03 €	4 Ud.	16,12 €
Tarjeta microSD 32 GB	6,98 €	1 Ud.	6,98 €
Arduino micro	18,00 €	1 Ud.	18,00 €
Batería portátil 20.000 mAh	18,00 €	1 Ud.	18,00 €
<b>TOTAL</b>			<b>155.79 €</b>

Tabla 4 Presupuesto mando de control.

### 3.4.2 Cámara inteligente

Producto	Precio unitario	Unidades	Precio total
Raspberry Pi 3 B+	35,65 €	1 Ud.	35,65 €
Modulo cámara Raspberry PI V 2.1	22,27 €	1 Ud.	22,27 €
Tarjeta microSD 64 GB	11,98 €	1 Ud.	11,98 €
<b>TOTAL</b>			<b>69.90 €</b>

Tabla 5 Presupuesto servidor-cámara inteligente

---

## CAPÍTULO 4

### DISEÑO DE LA SOLUCIÓN

---

En este capítulo, haré hincapié sobre la arquitectura en la cual se basa la aplicación y, las funciones que realizarán sus componentes. Asimismo, describiré e ilustraré los diagramas de flujo, que representan la lógica de la solución propuesta.

#### 4.1 ARQUITECTURA DEL SISTEMA

El sistema propuesto es una aplicación con arquitectura cliente-servidor. Este tipo de solución es un modelo de diseño de software donde su propósito es mantener comunicaciones de información entre diferentes entidades de una red; dicho modelo se caracteriza por dos partes claramente distinguibles:

Por un lado, el servidor, que en nuestro caso es Raspberry Pi, dotado de cámara e instalado en las distintas cintas de producción. Por otro lado, el cliente que es nuestro mando de control, formado por una ulterior Raspberry Pi y el Arduino Micro. El cliente es la parte demandante, realiza peticiones al proveedor, el servidor, quien le da la respuesta.

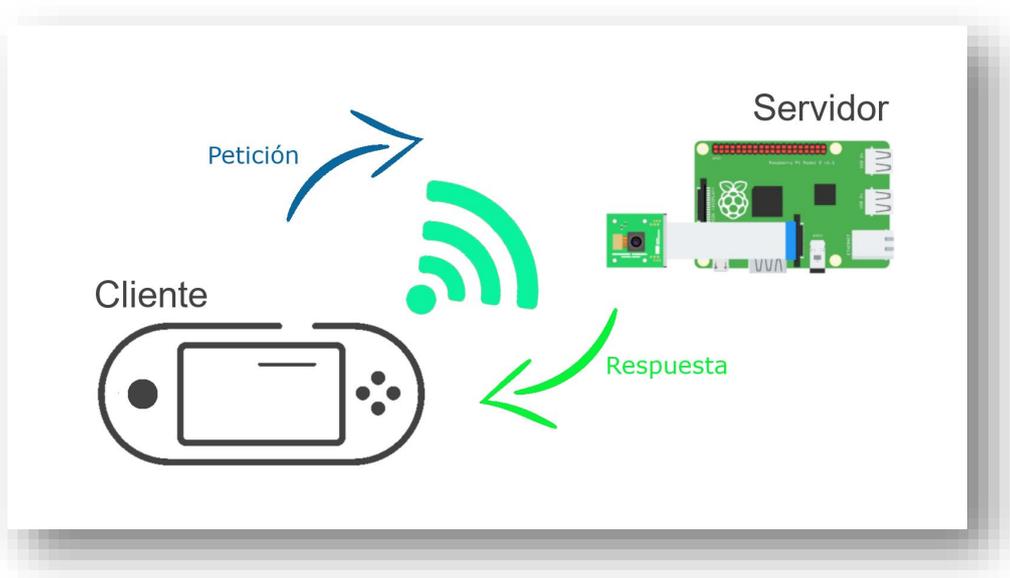


Figura 4-1. Arquitectura sistema cliente-servidor.

Cuando el cliente se conecta al servidor, automáticamente éste reconoce la conexión como una petición de inicializar la modalidad configuración, por lo tanto, se empieza la transmisión del vídeo en tiempo real enviando los fotogramas que capta la cámara.

Adicionalmente, el servidor se pone a la escucha de posibles mensajes que le son enviados por el cliente; estos mensajes en formato JSON contienen las nuevas configuraciones que debe adoptar la cámara. Si un mensaje es recibido por el servidor se adopta la nueva configuración y se reinician las operaciones.

Si no se encuentra ningún cliente conectado tras la inicialización del servidor, éste procede a la modalidad operativa analizando cada fotograma; primero comprueba si se ha verificado movimiento en la imagen, de ser así se procede a detectar la bandeja que contiene los productos.

En caso afirmativo, se procede a la búsqueda y conteo de los productos, y tras obtener la cantidad se comprueba que éste sea mayor o igual al umbral establecido. De cumplirse este requisito se procede a reiniciar el proceso, sin embargo, si no se cumple se asume que hay un error en la cadena de producción y por lo tanto, se guarda en memoria la fotografía representativa del error junto a su información.

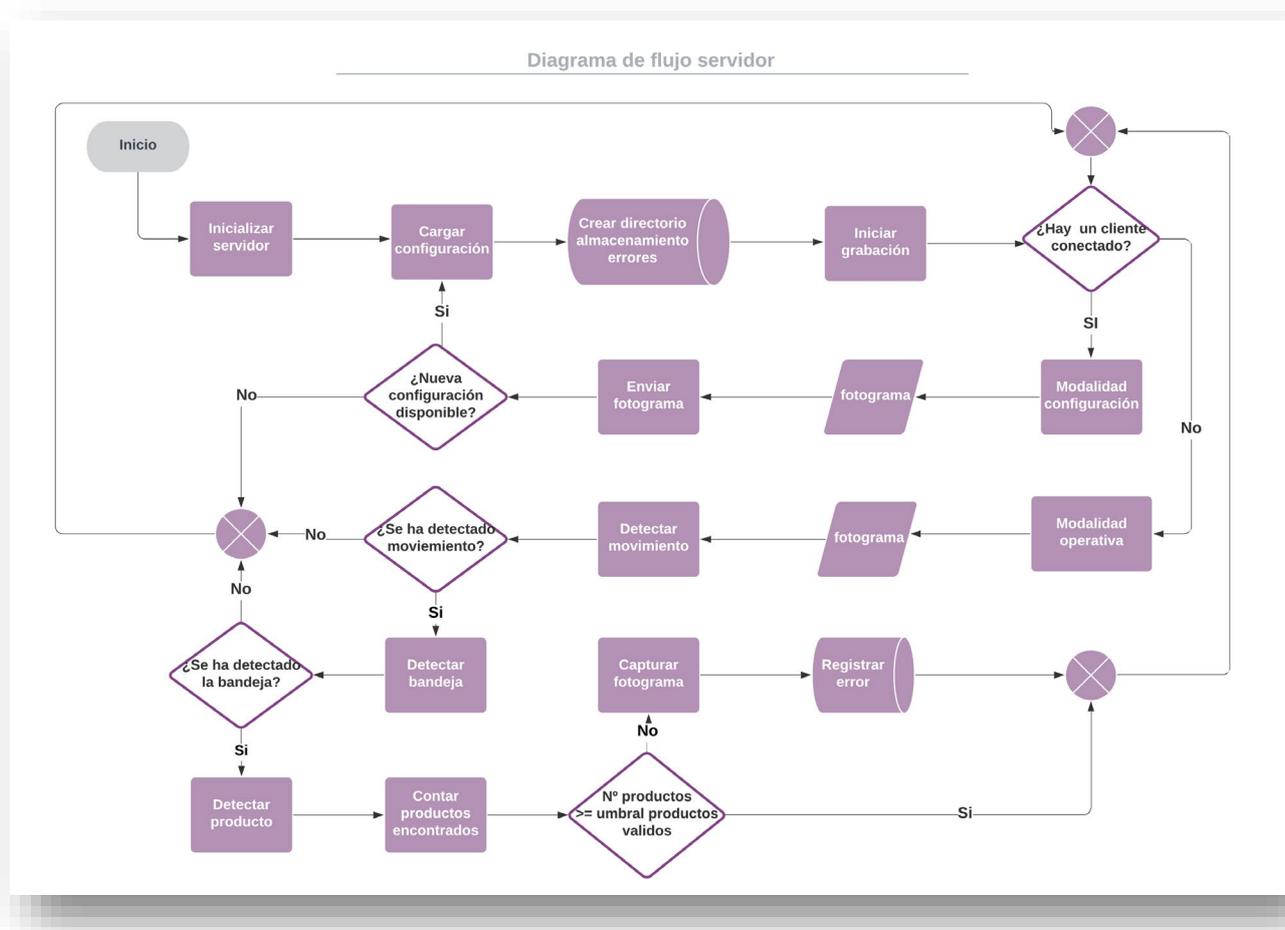


Figura 4-2. Diagrama de flujo lógica servidor.

Por otro lado, al inicializar el cliente, se muestra la ventana de conexión desde la cual el usuario puede conectarse al servidor después de haber digitado los parámetros de conexión, que en específico son, la dirección IP y el puerto.

El cliente, antes de intentar la conexión, comprueba que los datos insertados sean sintácticamente correctos y, sólo si superan el control, se inicia la conexión intentando conectarse al servidor.

En caso afirmativo, se pasa a la ventana de configuración, mientras que en caso de rechazo de la conexión se muestra un mensaje de error, describiendo el fallo causante de dicho rechazo.

En la ventana de configuración el usuario puede configurar la cámara y, una vez satisfecho de los cambios realizados, puede enviar la nueva configuración para que ésta sea adoptada por el servidor. Adicionalmente puede limitarse a observar la situación en tiempo real.

Si en cualquier momento dado se verifica una interrupción de la conexión, el cliente volverá a la pantalla de conexión explicando al usuario a través de un mensaje en la ventana, cuál ha sido el fallo que ha impedido el correcto funcionamiento de la aplicación.

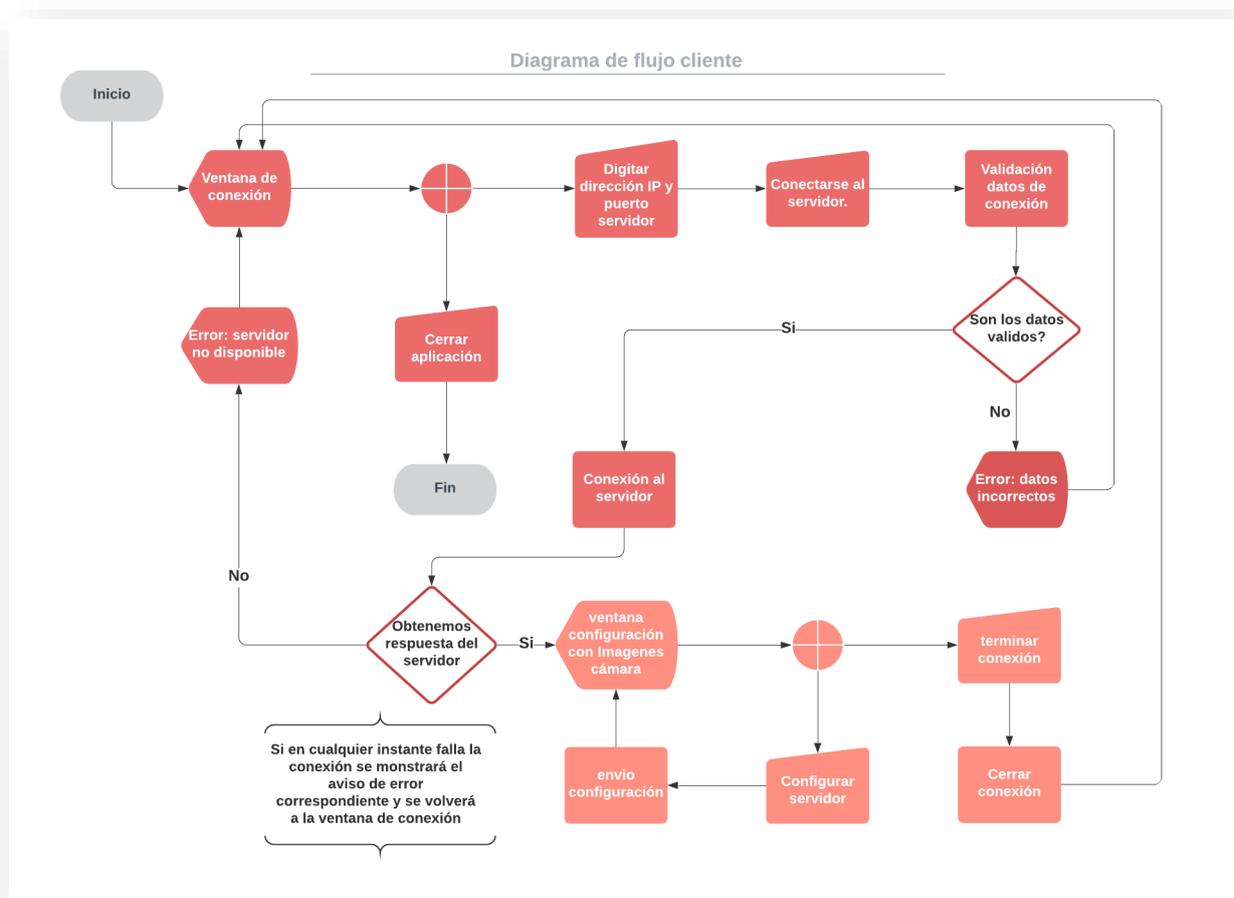


Figura 4-3. Diagrama de flujo lógica cliente.



## 4.2 TECNOLOGÍAS EMPLEADAS

Para llevar a cabo el proyecto, se utilizarán numerosas y diferentes tecnologías y componentes, tanto en la división hardware como en la software. A continuación, cito las más relevantes junto a sus principales características.

### 4.2.1 Hardware

En el sector hardware, se van a describir los principales componentes utilizados para la cámara inteligente, que a su vez se divide en la cámara propiamente dicha y el ordenador que la controla. De la misma manera, se comentan los componentes utilizados para construir el mando de control y las funciones que desempeñan en éste.

#### 4.2.1.1 Raspberry Pi 3 modelo B+

Como ya he comentado, el proyecto se basa en los ordenadores Raspberry pi, ya que sus pequeñas dimensiones, unidas a su bajo coste y sus considerables prestaciones, hacen que esta tipología de computadores sea perfecta para este tipo de aplicaciones.

Los ordenadores serán instalados, uno en el mando de control, y uno en la cámara inteligente; en ambos se instalará el software que permitirá el funcionamiento de los componentes.

Se empleará para el proyecto el modelo B+ de la familia Raspberry PI 3, del cual he citado antes sus características y que podemos observar en la siguiente imagen.



Figura 4-4. Raspberry Pi 3 modelo B+. Imagen extraída de [9].

#### 4.2.1.2 Arduino Micro

Arduino es una compañía de desarrollo de software y hardware libres, así como una plataforma de desarrollo de electrónica de código abierto, basada en una placa electrónica que incorpora un microcontrolador reprogramable y una serie de pines GPIO.[\[12\]](#)



Figura 4-5. Logo y productos Arduino. Imagen extraída de [\[13\]](#).

La placa electrónica en la cual se basa Arduino es una común PCB, (Placa de Circuito Impreso) que implementa un determinado diseño de circuitería interna, ésta es la forma más compacta y estable de construir un circuito electrónico.

De esta forma el usuario final no debe preocuparse por las conexiones eléctricas que necesite el microcontrolador para funcionar, y puede directamente centrarse en desarrollar las diferentes aplicaciones que desee.[\[12\]](#)

El microcontrolador presente en la placa electrónica es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria; en Arduino disponemos de una interfaz de entrada, los GPIO, éstos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

Las conexiones se realizan habitualmente con cables Dupont, y la información de los periféricos que son conectados a través de ellos se traslada al microcontrolador, el cual se encarga de procesar los datos que le lleguen mediante los mismos.[\[14\]](#)

Ésta es principalmente la función que realizará nuestro Arduino en el proyecto, se encargará de recibir la información de los pulsadores y del joystick instalados en el

mando de control, convertir las señales de analógico a digital cuando sea necesario, procesar los datos y enviarlos a la Raspberry Pi.

Los datos serán enviados a la Raspberry Pi a través de la interfaz de salida, que es la encargada de llevar la información procesada en el Arduino a otros periféricos. En nuestro Arduino dicha interfaz es el puerto micro USB.

¿Por qué se ha elegido Arduino frente a otros microcontroladores?

Principalmente se ha escogido Arduino por las siguientes motivaciones :

- Documentación extensa y fiable.
- Amplia comunidad trabajando con la plataforma.
- Lenguaje de programación de fácil comprensión (basado en C++).
- Bajo coste.
- Reusabilidad y versatilidad.

El modelo así seleccionado, principalmente por sus dimensiones compactas, necesarias para ocupar el menor espacio posible en el interior del mando de control, y por poseer las componentes básicas necesarias al proyecto es, el Arduino Micro, que podemos observar en la siguiente figura.

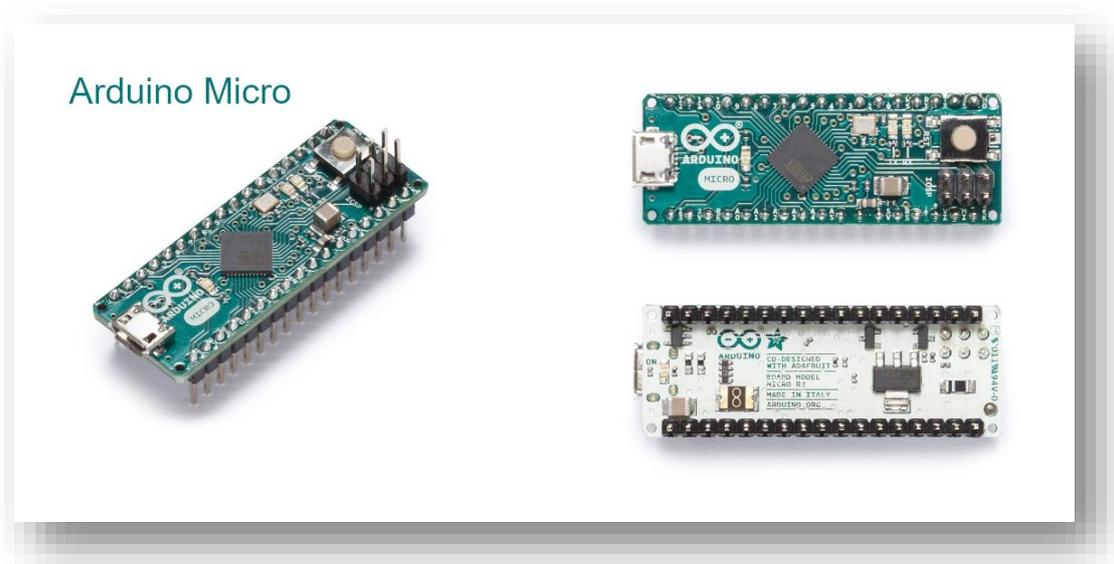


Figura 4-6. Arduino Micro. Imagen extraída de [13]

Arduino Micro es una placa con microcontrolador basado en el ATmega32U4, posee 20 pines configurables como entradas o salidas, de los cuales 7 se pueden usar como salidas PWM y 12 como entradas analógicas, conexión micro USB y botón de reinicio.[13]

<b>Especificaciones técnicas Arduino Micro</b>	
<b>Microcontrolador</b>	ATmega32U4
<b>Voltaje de funcionamiento</b>	5 V
<b>Voltaje de entrada recomendado</b>	7-12 V
<b>Voltaje de entrada limite</b>	6-20 V
<b>Pines I/O digitales</b>	6-20 V
<b>Canales PWM</b>	7
<b>Canales analógicos de entrada</b>	12
<b>DC por pin I/O</b>	20 mA
<b>DC por pin de 3.3V</b>	50 mA
<b>Memoria Flash</b>	32 KB (ATmega32U4)
<b>SRAMM</b>	2,5 KB (ATmega32U4)
<b>Frecuencia de reloj</b>	16 MHz
<b>Anchura</b>	18 mm
<b>Longitud</b>	48 mm
<b>Peso</b>	13 gr

Tabla 6 Especificaciones técnicas Arduino Micro. Tabla extraída de [13].

#### 4.2.1.3 Pi Camera modulo v2

La cámara inteligente, además de contar para el componente computador con la mencionada anteriormente Raspberry Pi, contará con una Pi camera modulo v2 que representará sus componentes cámara y unidad óptica.

La Pi camera modulo v2, es como sugiere su nombre, un módulo de cámara óptica fabricada por Raspberry Pi de fácil acoplamiento para sus dispositivos, y puede servir tanto para realizar grabaciones en alta definición como para tomar fotografías.

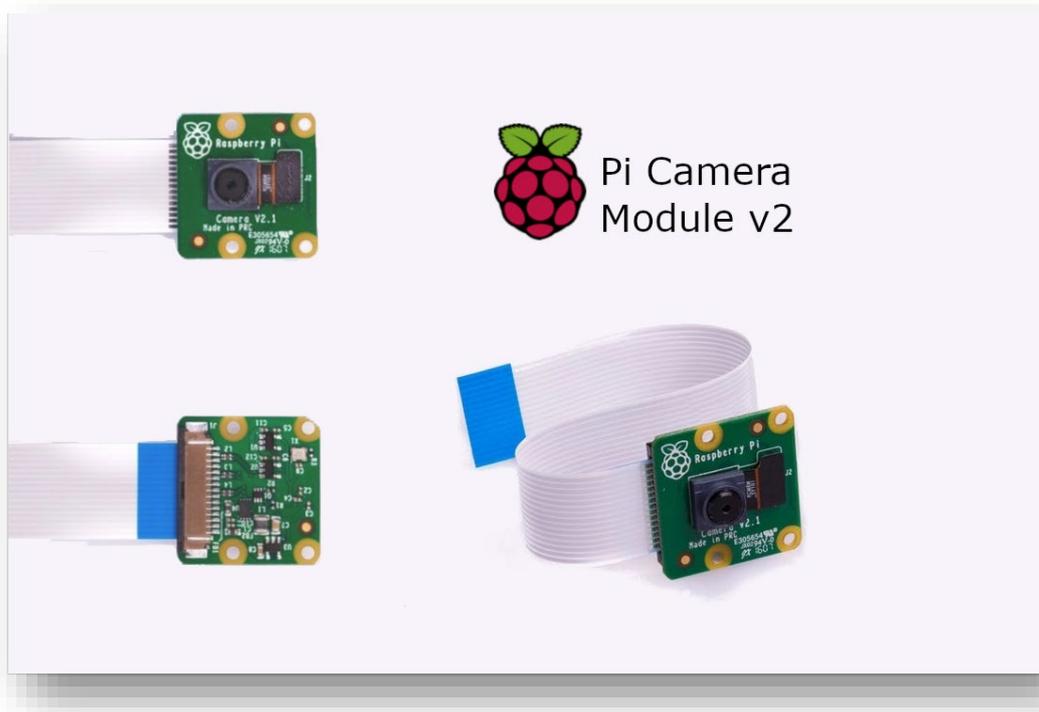


Figura 4-7. Pi camera module v2. Imagen extraída de [14].

Incluye un sensor IMX219 de la marca Sony de 8-megapíxeles que nos permite realizar grabaciones, entre otras configuraciones, a 1080 píxeles a una frecuencia de 30 fps. o, a 720 píxeles con una frecuencia de 60 fps.

Se ha elegido específicamente este componente, entre otros, por las siguientes motivaciones:

- **Compatibilidad.**  
Dicho modulo es compatible con todos los modelos de Raspberry Pi 1,2,3 y 4. De esta manera, en las posibles futuras actualizaciones del ordenador no será necesario sustituir también la componente cámara.
- **Usabilidad.**  
Esta cámara posee una gran usabilidad, es perfecta para los principiantes por su facilidad de manejo, y al mismo tiempo dispone de una gran variedad de características para los usuarios más expertos.
- **Documentación.**  
Al igual que los ordenadores, los módulos de cámara Raspberry Pi poseen una extensa comunidad que redacta y provee una vasta cantidad de documentación sobre su función, librerías de códigos y proyectos.

- Dimensiones

Sus dimensiones, extremadamente compactas, y su peso reducido, hacen que pueda ser ubicada en cualquier emplazamiento deseado sin necesidad de instalaciones complejas.

Especificaciones técnicas Pi camera module v2	
<b>Sensor óptico</b>	Sony IMX219
<b>Resolución</b>	8 MP
<b>Resolución fotografías</b>	3280x2464
<b>Resolución grabaciones</b>	<ul style="list-style-type: none"> <li>• 1080 p, 30 fps.</li> <li>• 720 p, 60 fps.</li> <li>• 640 x 480 p, 90 fps.</li> </ul>
<b>Distancia focal</b>	fija: 3,04 mm
<b>Tipología bus</b>	MIPI, CSI
<b>Formato lente</b>	1/4 "
<b>Apertura</b>	2.0 F
<b>Angulo de visión</b>	62.2 °
<b>Espesor</b>	9 mm
<b>Anchura</b>	25 mm
<b>Longitud</b>	23 mm
<b>Peso</b>	3 gr

Tabla 7. Especificaciones técnicas Pi camera module v2.

#### 4.2.1.4 Pantalla Raspberry Pi Touch

En el mando de control, con el objetivo de poder ofrecer al usuario final la posibilidad de ver en tiempo real, lo que está ocurriendo en las determinadas líneas e interactuar con la cámara, es necesario instalar una pantalla táctil.

Para ello, se ha optado por la pantalla táctil oficial de Raspberry Pi, ya que ofrece preinstalados los soportes para poder acoplar el ordenador, y es compatible con la totalidad de los dispositivos Pi.

Su conexión es realmente sencilla pues se conecta a Raspberry mediante un adaptador que se encarga de controlar la alimentación y la señal de vídeo. En éste se enlazan la alimentación mencionada anteriormente, proveniente desde el conector GPIO y el cable plano desde el conector DSI del ordenador.[\[16\]](#)



Figura 4-8. Raspberry Pi Touch. Imagen extraída de [17]

La pantalla, con una resolución de 800 x 480 píxeles ofrece 10 puntos de presión simultáneos, facilitando el uso de teclados en pantalla. A continuación, se muestran las restantes especificaciones técnicas.

Especificaciones técnicas Pi touch	
Resolución pantalla	800 x 480 píxeles
Tamaño pantalla	7 pulgadas
Interfaz	DSI
Alimentación	GPIO
Sistema operativo	Raspian
Puntos de presión	10
Espesor	20 mm
Anchura	110 mm
Longitud	194 mm
Peso	0,44 kg

Tabla 8. Especificaciones técnicas Pi touch.

## 4.2.2 Software

En la división de software, haré hincapié sobre el entorno del desarrollo escogido para realizar la aplicación y los lenguajes de programación. Adicionalmente, se mencionarán otros programas que han ayudado en el curso de la solución.

### 4.2.2.1 Processing

Processing es, un lenguaje de programación y entorno flexible, de desarrollo integrado de código abierto, basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.[\[18\]](#)

Al estar basado en Java hereda toda su funcionalidad, pero a pesar de ello, también presenta versiones en Python y para programación en Android. Su potencial es extenso gracias a la presencia de numerosas librerías y herramientas ofrecidas por la comunidad de Processing.

Se cuentan más de 100 librerías existentes; éstas pueden ser englobadas en el desarrollo de una aplicación con el fin de permitirnos nuevas funcionalidades, como pueden ser trabajar con geometrías 3D avanzadas, realizar sistemas de visión por computador (nuestro caso), e instrumentos para la manipulación y creación de sonidos.[\[19\]](#)

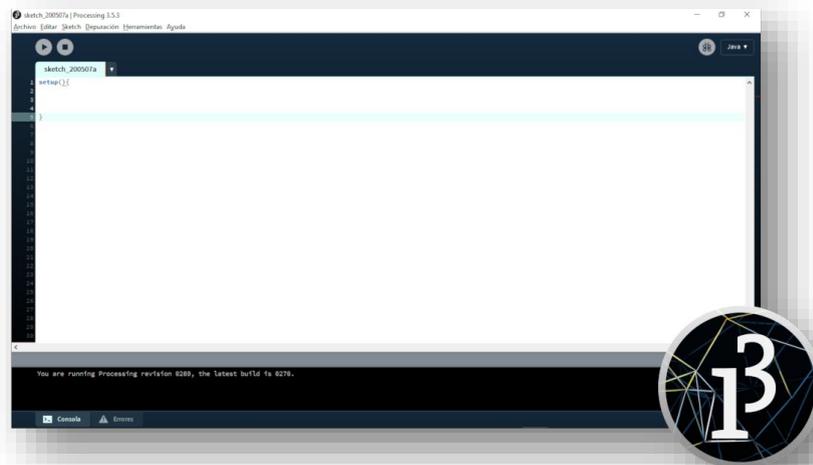


Figura 4-9. Entorno de desarrollo Processing con Logo.

En este contexto, un programa se llama sketch (dibujo). Los dibujos son almacenados en el sketchbook (libro de dibujos), que es simplemente una carpeta en nuestro ordenador. Dichos programas pueden renderizar gráficas en dos y tres dimensiones, lo que hace posible en el caso de las tres dimensiones, el controlar la cámara, la luminosidad y los materiales.

Ambos programas, tanto el software del mando de control como el de la cámara inteligente, se realizan en este ámbito de desarrollo. Se ha seleccionado esta tecnología ya que presenta numerosas facilidades a la hora de trabajar con elementos gráficos en sistemas de visión por ordenador.

Asimismo, las aplicaciones pueden ser exportadas y ejecutadas en numerosos sistemas operativos, entre los cuales podemos encontrar los conocidos Microsoft, Mac OS, Linux..., incluyendo además una gran cantidad de variantes menos conocidas de Linux para arquitecturas ARM.

#### 4.2.2.2 *Arduino IDE*

El software del microcontrolador se realizará en el entorno de desarrollo integrado de Arduino: Arduino IDE.

El entorno es una aplicación multiplataforma, escrita en el lenguaje de programación Java y basada en Processing y otros programas libres. Puede ser usado con todas las placas existentes de Arduino, pero también con placas de desarrollo de otros proveedores.[\[20\]](#)

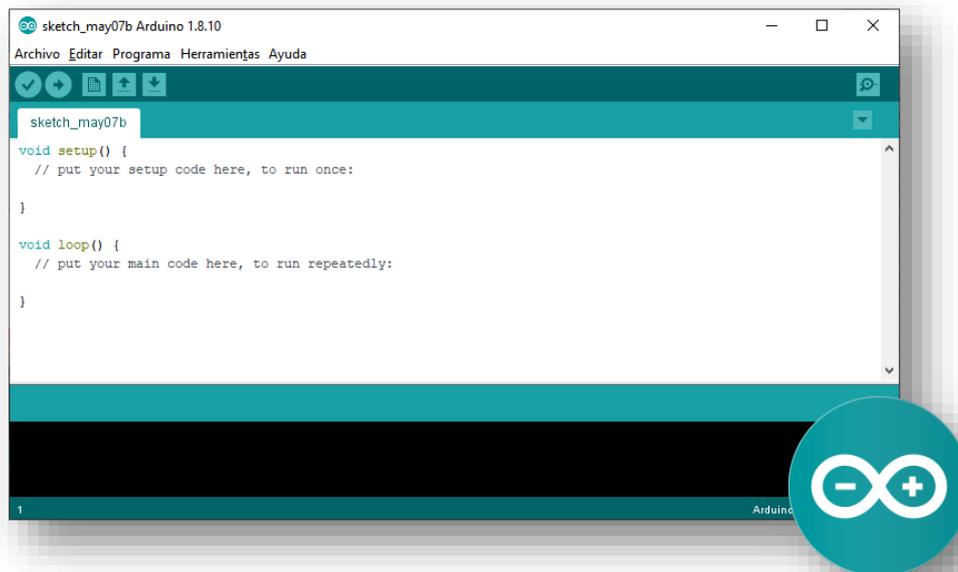


Figura 4-10. Entorno de desarrollo Arduino IDE con logo.

Este software libre, permite desarrollar fácilmente el código y subirlo a la placa, y puede ser ejecutado en los principales sistemas operativos Windows, Mac OS y Linux. Adicionalmente posee una versión on-line desde la cual podemos realizar nuestras aplicaciones sin realizar ninguna instalación.[\[20\]](#)

El IDE admite los lenguajes C o C++ utilizando reglas especiales de estructuración de códigos. Dicho código, escrito por el usuario, sólo requiere de dos funciones básicas, una para iniciar la aplicación llamada setup, y otra para ciclo principal del programa llamada loop, que se compilan y vinculan con un apéndice del programa principal.[\[21\]](#)

#### 4.2.2.3 OpenCV

OpenCV, que significa *Open Computer Vision* (Visión Artificial Abierta), es una librería de visión por ordenador y aprendizaje automático, de código abierto, originariamente desarrollada por Intel. Su principal objetivo es el de ofrecer una infraestructura común para todas las aplicaciones de visión artificial.[\[22\]](#)

Se ha utilizado en una gran cantidad de aplicaciones, y en la actualidad se sigue mencionando como la biblioteca más popular de visión artificial. Ello es debido a que OpenCv viene exhibida bajo licencia BSD (Distribución Software Berkeley), lo que permite que sea usada libremente para propósitos comerciales y de investigación.[\[23\]](#)

Adicionalmente, puede ser utilizada en diferentes plataformas, siendo disponible para los sistemas operativos GNU/Linux, Mac OS X, Windows y Android, y para diversas arquitecturas de hardware, entre ellas ARM.[\[23\]](#)

A través de los más de 2.500 algoritmos optimizados que incluye la librería, podemos detectar y reconocer caras, identificar objetos, clasificar acciones humanas, trazar movimientos de objetos y de cámaras, extraer modelos 3D y mucho más.[\[22\]](#)

Todas estas características, junto a una extensa y actualizada documentación disponible por parte de la organización OpenCV, tanto para desarrolladores novatos en el campo de la visión artificial como profesionales, han decantado su elección para el proyecto.

#### 4.2.2.4 OpenGL

OpenGL es una especificación estándar que define una API multilenguaje y multiplataforma, nacida en 1992, para desarrollar aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes, que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. [\[24\]](#)

Desde su introducción se ha convertido en la más utilizada y admitida en la industria dentro de su ámbito, debido a una serie de ventajas impulsadas por el desarrollador; entre las más importantes se encuentran: [\[25\]](#)

- Estándar de la industria: Un consorcio independiente guía las especificaciones de OpenGL ofreciendo un amplio soporte de la industria.[\[25\]](#)



- Fiable y portátil: Todas las aplicaciones OpenGL producen resultados consistentes en cualquier hardware compatible, independientemente del sistema operativo o del sistema de ventanas.[\[25\]](#)
- Documentación: “Se han publicado numerosos libros sobre OpenGL, y una gran cantidad de código de muestra está fácilmente disponible, lo que hace que la información sobre OpenGL sea económica y fácil de obtener.”[\[25\]](#)

Ya que, en el proyecto, un papel fundamental será ejercitado por la captura y envío de imágenes, y adicionalmente se utilizará ampliamente la edición sobre imagen de rectas y figuras geométricas, tanto para las áreas de detección de movimiento como para las de detección de perímetro, se ha decidido utilizar OpenGL por sus características y funciones.

#### **4.2.2.5 ControlP5**

ControlP5 es el nombre de una librería desarrollada por Andreas Schlegel para el entorno de programación Processing, cuya función es facilitar la creación de interfaces gráficas de usuarios.[\[26\]](#)

Dicha librería incluye los controladores de los varios elementos gráficos disponibles; éstos permiten desencadenar las acciones deseadas por el usuario, cuando este último interactúa con ellos. Entre sus principales componentes encontramos: botones, gráficos, campos de texto, y tablas que se pueden organizar en controladores diferentes o agrupar por pestañas.[\[26\]](#)

Se ha elegido esta librería para realizar la interfaz gráfica del usuario en el mando de control, debido a su gran cantidad de características y a su compatibilidad con pantallas táctiles y convencionales.

#### **4.2.2.6 OpenCV-Processing**

OpenCV-Processing es una librería de visión por computador basada en la versión oficial de OpenCV para Java, donde su objetivo es el de ofrecer un wrapper para los métodos más utilizados por esta última, de manera que se simplifique su usabilidad para los desarrolladores principiantes y que sean familiares con el entorno de desarrollo de Processing.[\[27\]](#)

Adicionalmente, ofrece la posibilidad de importar directamente los módulos de OpenCV sin utilizar el wrapper que provee esta librería y utilizar cualquiera de las funciones documentadas.

Se ha elegido emplear esta librería y sus métodos además de utilizar la oficial de OpenCV, ya que es una adaptación simplificada para el entorno de Processing con una extensa documentación y una gran cantidad de ejemplos disponibles.



---

---

# CAPÍTULO 5

## DESARROLLO DE LA SOLUCIÓN PROPUESTA

---

A continuación, se va a exponer el desarrollo de la solución propuesta, para ello se procede a analizar y explicar cada una de las partes que la componen: el servidor y el mando de control. Dentro de dichos componentes, se van a encontrar dos secciones principales que representan los diferentes ámbitos dentro de los cuales se ha desarrollado la aplicación: Hardware y Software.

En cada apartado, se hace hincapié sobre las principales funcionalidades presentes, cómo se han conseguido, los problemas relevantes encontrados en el desarrollo y por último su rol dentro del sistema.

### 5.1 MANDO DE CONTROL

El mando de control ejerce la función de “cliente” dentro de nuestra arquitectura cliente-servidor; mediante éste, el usuario podrá conectarse a las varias cámaras instaladas en las diferentes líneas, ver en directo lo que está ocurriendo y configurarlas.

A continuación, dentro de la sección hardware se van a ilustrar los componentes empleados en su construcción con el rol que desempeñan, las conexiones entre ellos a través de esquemas ilustrativos y el ensamblaje final.

En la sección software, se procede a exponer detalladamente el código empleado en el mando de control, explicando sus funciones y el fin para el cual se ha desarrollado.

#### 5.1.1 Hardware

El mando de control tiene la forma de una consola portátil para facilitar la interacción con el usuario; la carcasa que le da dicha forma está enteramente construida a través de impresión 3D. En el centro encontramos ubicada la pantalla, mientras que a los lados de ésta se encuentran los pulsadores y el joystick.



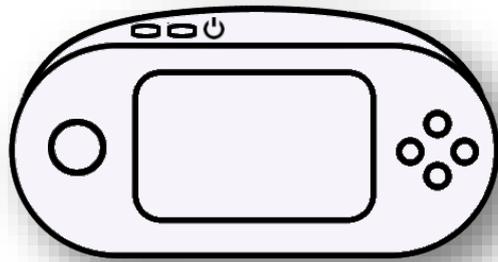


Figura 5-1. Mockup del mando de control.

En la parte superior se encuentra el botón de encendido; adicionalmente, son visibles dos aberturas con conectores tipo USB que conectan respectivamente la batería, para poder realizar su carga, y un puerto de la Raspberry PI para poder conectarse a ella.

En el interior encontramos, además de los componentes descritos anteriormente, el Arduino micro y el cableado. El Arduino micro desempeña el papel de controlar, decodificar y gestionar las señales que le llegan por parte de los pulsadores antes de enviarlas al ordenador.



Figura 5-2. Elementos hardware que componen el mando de control.

Para ello se conectan las salidas del joystick soldando los cables a los GPIO del microcontrolador. En el joystick seleccionado disponemos de dos salidas analógicas para los valores del eje X y del eje Y, una salida digital que indica cuándo es presionado (selección), alimentación de 5 V y toma de tierra.

Cuando el joystick es pulsado se presiona el botón integrado de selección y éste envía un valor correspondiente a tierra (0) que nos dice que el botón ha sido pulsado. Dicha salida debe ser conectada a un pin digital dada la tipología de su valor.

Como mencionado anteriormente las salidas X e Y son analógicas; éstas actúan como unas resistencias variables de 10 K y ambas nos proporcionan valores en una escala de 0 a 1.023, donde 0 es el valor mínimo y 1.023 es el máximo.

El joystick sigue el eje cartesiano y su sistema de coordenadas, formado por las tuplas de valores X e Y. El centro oscila entre los valores 507-509 para la X y 508-510 para la Y; en estas coordenadas el joystick se encuentra en estado de reposo.

Esto es debido a que el centro se encuentra en la mitad de la escala de valores, y al ser señales analógicas no tenemos una exactitud absoluta, sino un rango dentro del cual los valores oscilan.

En la siguiente imagen, se pueden ver los valores que nos proporciona el joystick si éste es movido a los extremos del eje, dichos extremos son representados por los puntos de color azul; en efecto si se mueve el joystick al origen obtenemos los valores (0, 0).

Si lo desplazamos desde la posición de reposo hacia arriba, obtendremos unos valores para la coordenada Y que van desde 510 con tope en 1.023, si lo movemos hacia abajo nos dará unos valores desde 508 con limite en 0.

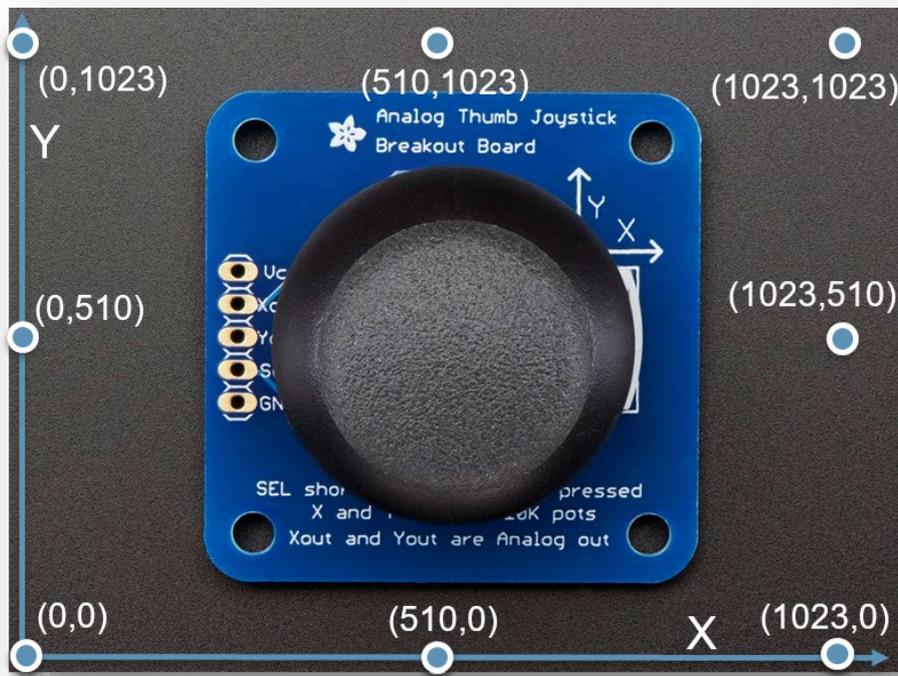


Figura 5-3. Valores analógicos joystick en los ejes cartesianos.

Los pulsadores elegidos actúan como resistencias pull up, estableciendo un estado alto (5 V) cuando el botón no es pulsado, lo que equivale al valor 1 lógico. Sin embargo, si el botón es pulsado se establece un estado bajo.

Las patillas de los pulsadores son conectadas en un extremo con un pin digital del Arduino Micro y en el otro extremo con la masa (GND), cuando son apretados éstos enviarán un 0 al pin digital al cual están conectados.

El usuario, utilizará estos botones para seleccionar un determinado lado del rectángulo que delimita las áreas de detección de movimiento o de perímetro y poder editarlos según desee. En efecto, el pulsador a la derecha corresponde con el lado derecho, el inferior con el lado inferior, y así los restantes.

Una vez realizadas todas las conexiones entre los varios componentes y el Arduino Micro, obtendremos como resultado la representación del siguiente esquema.

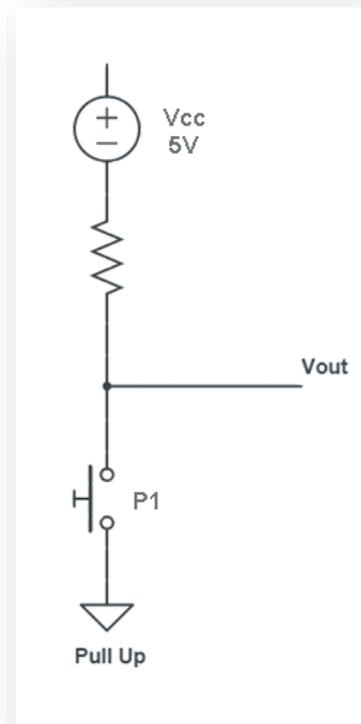


Figura 5-4. Esquema resistencia pull up. Imagen extraída de [28].

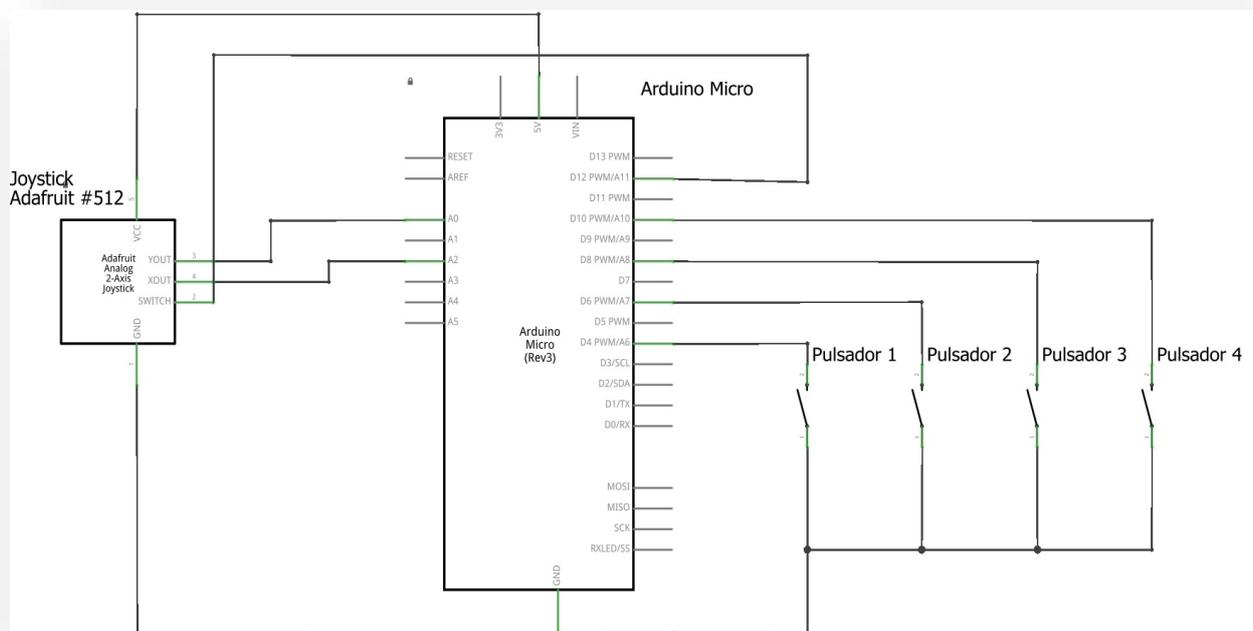
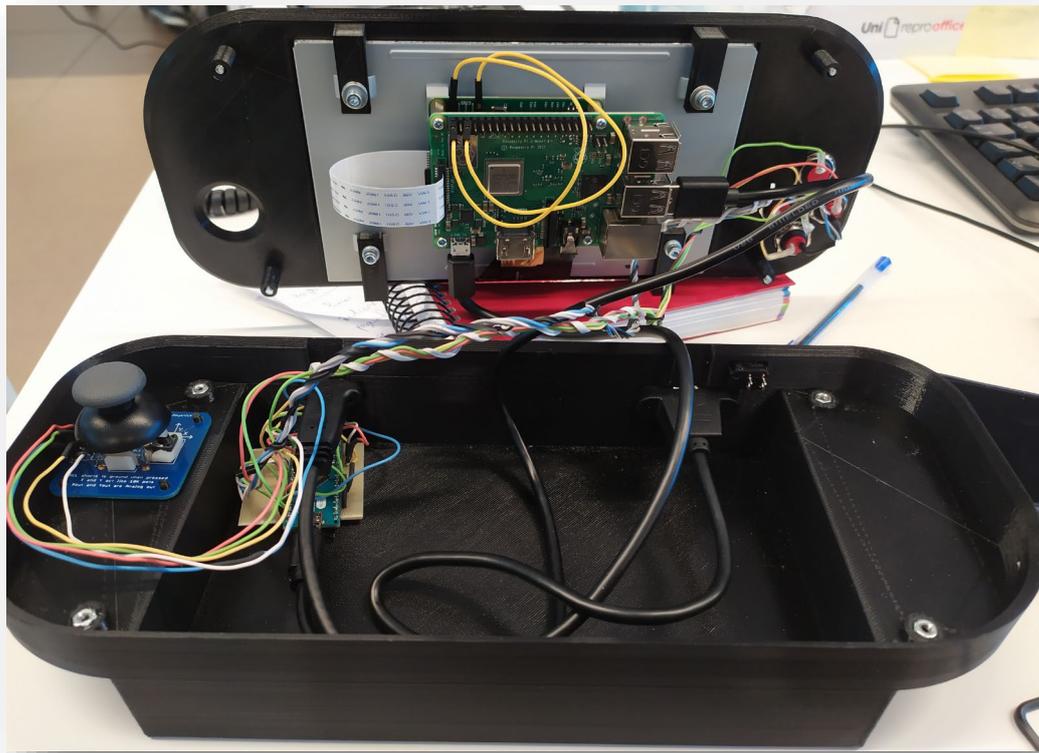


Figura 5-5 Esquema conexiones Arduino Micro.



Cabe destacar que los pines digitales y analógicos, pueden diferir de los que se ven en el esquema, ya que podemos elegir entre varios GPIO. Solamente es necesario respetar la tipología de la salida, es decir, las salidas digitales deben ser conectadas a pines digitales mientras que las analógicas a pines de esta tipología.

De igual manera, la alimentación del joystick es necesario que se conecte al GPIO de 5 V del Arduino Micro ya que es su voltaje de funcionamiento; si se conecta al pin de 3 V obtendremos resultados indeseados. En el apartado de anexos se pueden encontrar los varios GPIO disponibles en el Arduino Micro.



*Figura 5-6 Montaje componentes hardware en el mando de control.*

La conexión entre la Raspberry PI y el Arduino, como se puede ver en la imagen anterior, se realiza a través de un cable USB – micro USB. Además, se conecta el ordenador con la pantalla táctil mediante los cables de alimentación, que van desde los GPIO de la PI hasta la pantalla, y con el cable de vídeo que conecta los puertos DSI presentes.

A continuación, se conecta la batería portátil al ordenador y después se une la parte superior de la carcasa con la inferior que contiene la mayoría de los componentes y el cableado a través de unos tornillos ya incluidos. Tras estos pasos el mando de control está ultimado.



*Figura 5-7 Mando de control.*

### **5.1.2 Software**

En esta sección describiré qué funciones desempeñan los programas que son incluidos en el mando de control. Dicho mando presenta dos aplicaciones, una que se ejecuta en el Arduino mientras que la otra se ubica en el ordenador de placa reducida.

La aplicación presente en el Arduino se encarga de gestionar los datos que son enviados por el joystick y los pulsadores, extrapolar la información necesaria y enviarla al ordenador.

El software presente en el ordenador es el que interactúa con el usuario, a través de éste podemos conectarnos a las cámaras instaladas, observar en tiempo real la línea y configurar la cámara.

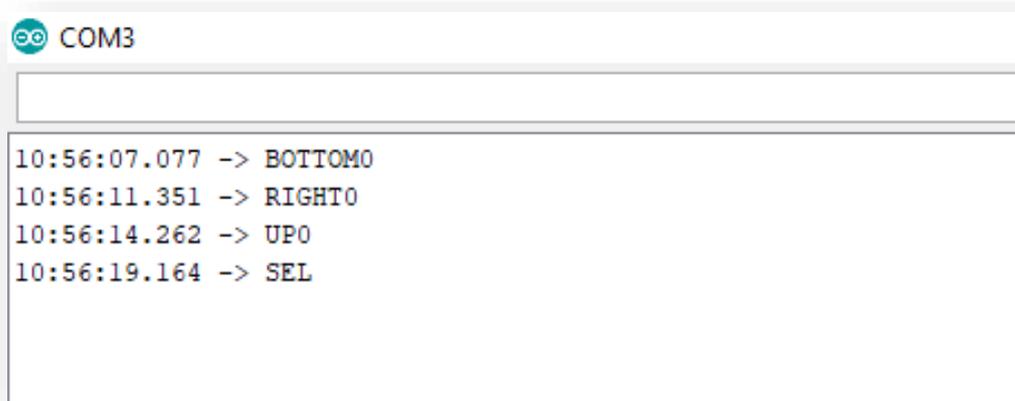
#### **5.1.2.1 Aplicación Arduino**

En primera instancia, se clasifican los pines digitales seleccionados como pines de entrada pull up. Dichos pines son los que están conectados a los pulsadores y al botón selección del joystick. Seguidamente se inicializa la comunicación por el puerto serie que servirá para enviar las acciones que debe realizar la aplicación en la Raspberry Pi.

A partir de este momento, el software se mantiene a la escucha de los datos que le llegan por los varios GPIO conectados, si en los pines de los pulsadores se registra un estado bajo, que es equivalente al 0 lógico, significa que uno de los botones ha sido pulsado.

Cada botón está asignado a un lado diferente del rectángulo que delimita el área de detección o de movimiento, por lo tanto, se envía la trama que constituye la información de que dicho lado ha sido seleccionado. La trama está formada por el nombre del lado seleccionado más el estado registrado.

En efecto, si es pulsado el botón superior se enviará al ordenador la trama "UP0", si es pulsado el botón izquierdo se enviará "LEFT0" y así para los restantes. El mismo procedimiento es seguido por el botón selección con la única diferencia que en este caso se envía la siguiente trama: "SEL".



```
COM3
10:56:07.077 -> BOTTOM0
10:56:11.351 -> RIGHT0
10:56:14.262 -> UP0
10:56:19.164 -> SEL
```

Figura 5-8 Datos enviados observados por el monitor del puerto serie.

Debido a posibles interferencias y a la acción propia de activación del comando selección, puede verificarse que lleguen señales de activación erróneas inmediatamente después a la correcta, haciendo que parezca que el botón ha sido pulsado múltiples veces.

Para evitar esta situación, se ha insertado una parada de 220 ms, donde dentro de este intervalo de tiempo el programa no acepta otras señales desde dicho botón. Dicho tiempo, no perceptible al usuario pero significativo para el software, se ha obtenido realizando pruebas físicas con el Arduino y el Joystick. Como resultado de dichas pruebas se ha constatado que con tiempos de parada superiores o iguales a 200 ms se garantiza la ausencia de falsos positivos.

Los pines analógicos, captan los valores de las coordenadas X e Y que proceden desde el joystick, para evitar enviar datos sin relevancia producidos por el joystick en reposo o por movimientos involuntarios, se han establecido unos umbrales, a partir de los cuales el dato de las coordenadas es considerado relevante y es tratado.

Cada vez que se recibe un dato válido se envía al ordenador la orden para mover la zona deseada con el desplazamiento a realizar; esta orden está compuesta por la coordenada X o Y, que indica el eje, el signo menos en el caso que la dirección sea contraria a la del eje y, el valor de cuanto se tiene que mover o modificar la zona.

Para conocer dicho valor y con el objetivo de ofrecer al usuario la posibilidad de editar a diferentes velocidades los objetos seleccionados, se han creado tres diferentes sectores dentro de la escala de valores que puede asumir la coordenada.

Dependiendo en que sector se encuentre el valor que es recibido por el pin, se enviará un valor dado de movimiento; cuanto más lejos el sector del estado de reposo, mayor será dicho valor. En concreto, si el valor de la coordenada pertenece al primer sector, su valor de movimiento correspondiente será 1 px, si pertenece al segundo 2 px y para el último sector 3 px.

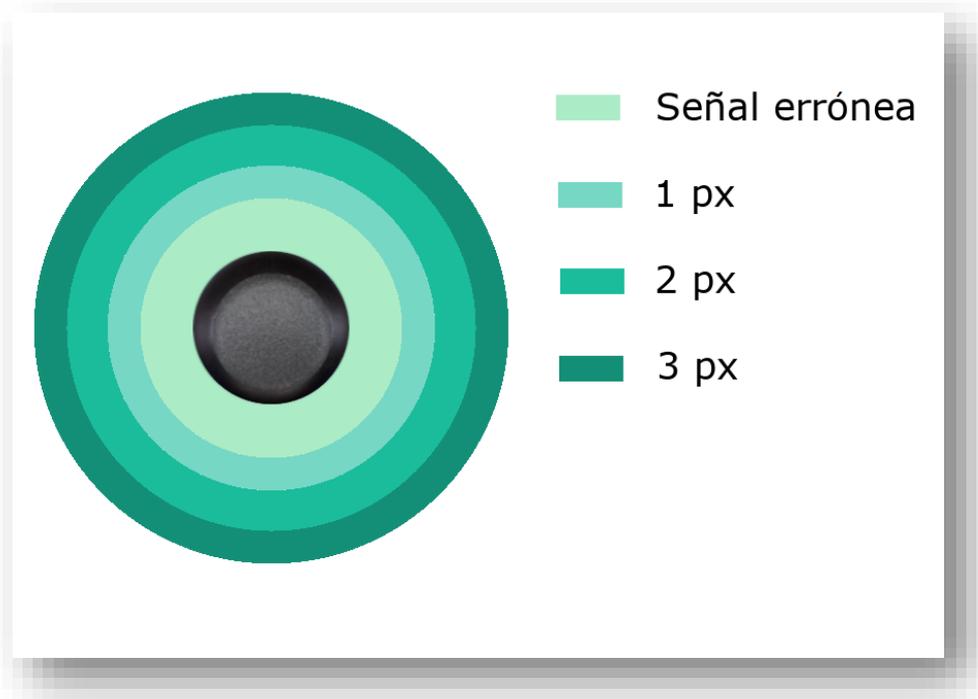


Figura 5-9. Correlación entre movimiento del joystick y valor enviado.

Como podemos apreciar en la figura anterior, estos valores indican los píxeles que se debe mover el objeto seleccionado; no debe engañar el valor extremadamente bajo ya que la frecuencia de envío es muy elevada. Por lo tanto, si se mueve el joystick en el eje X según su dirección hasta el tercer sector, se enviará al ordenador la siguiente orden: "X3". Sin embargo, si se mueve en la dirección contraria hasta llegar al mismo sector se enviará: "X-3".

Este mecanismo se aplica también a la coordenada Y; en caso de movimientos que afectan ambos ejes como puede ser un desplazamiento en diagonal, se enviarán dos tramas, una para el movimiento sobre el eje horizontal y otra correspondiente al eje vertical.

### 5.1.2.2 Aplicación Raspberry Pi

La aplicación se ejecuta en modo automático, cada vez que el ordenador es encendido, sin que el usuario tenga que realizar ninguna acción. Cabe destacar que, el software, es compatible también con ordenadores ordinarios y no sólo de placa reducida, por lo tanto, si en un momento dado se desea ejecutar esta aplicación desde otro terminal, dicho cambio se puede realizar sencillamente.

Para las interfaces de la aplicación, se ha elegido un estilo amigable, ligero, motivador y muy visual, cercano al usuario, ya que, según los análisis de requisitos efectuados, emerge como objetivo que, este software puede ser utilizado por trabajadores con tareas u ocupaciones, que no necesariamente se encuentren en el ámbito informático y, no tengan conocimientos en esta materia.

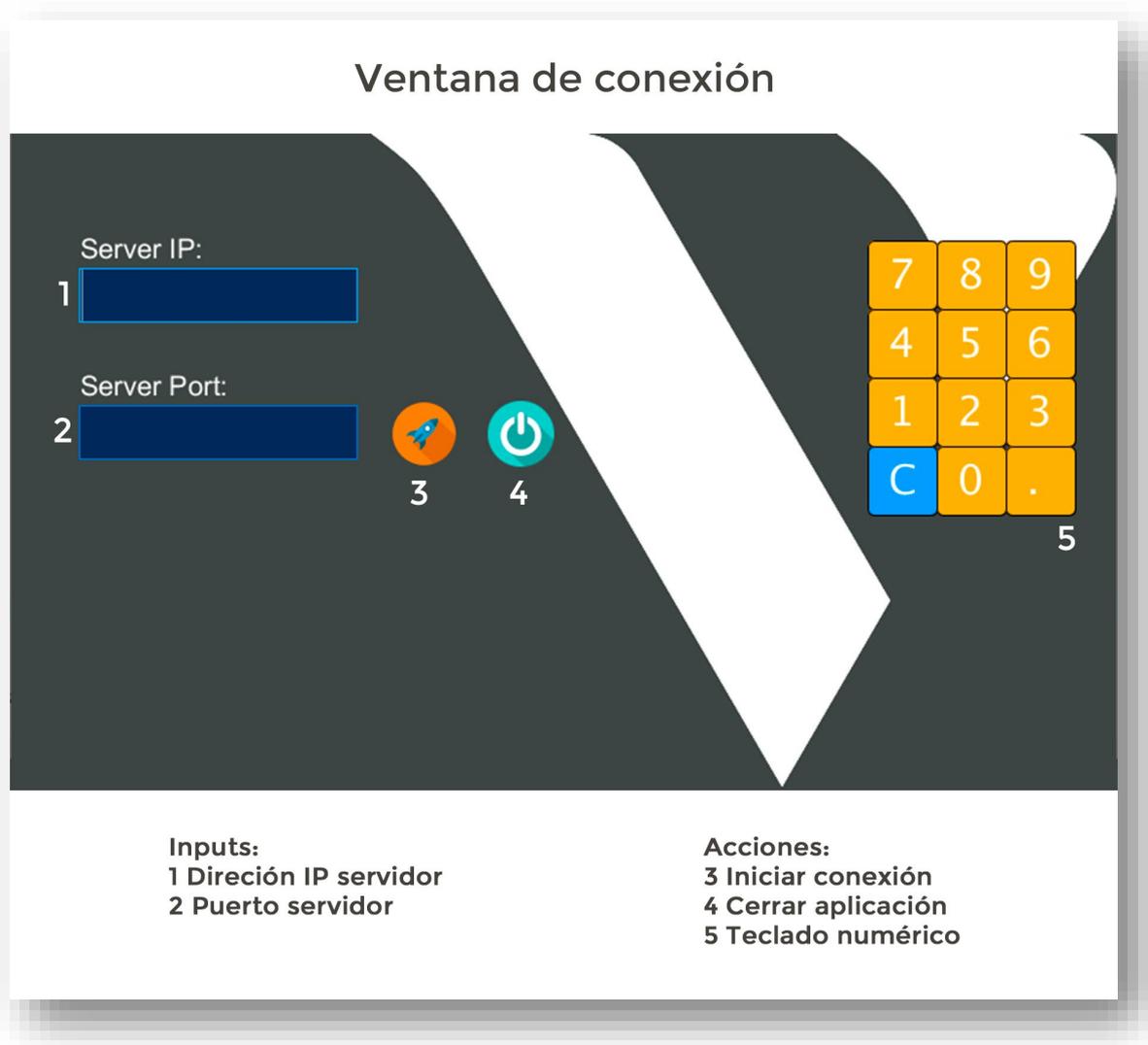


Figura 5-10 Ventana de conexión de la aplicación.

La ventana que se muestra al iniciar el software es la de conexión a las cámaras. En ella se encuentran dos campos de texto, que hacen referencia a la dirección IP del servidor y al puerto correspondiente, los botones para iniciar la conexión y apagar el sistema y, por último, un teclado numérico con el cual digitar las direcciones.

Antes de mostrar esta interfaz al usuario, el programa realiza unas comprobaciones internas para verificar la conexión con el Arduino a través del puerto serie. Si la conexión falla o el puerto serie no es detectado, aparece en la ventana un mensaje de error, avisando al usuario del problema e impidiéndole avanzar.

El mecanismo del teclado, difiere de los demás pulsadores presentes en la aplicación al no tener un controlador asociado que desencadene la acción deseada; cada vez que el usuario pulsa la pantalla táctil, se registran las coordenadas de su toque; si éstas se encuentran dentro del área correspondiente a una tecla dada, se escribe el valor asociado simulando la acción de apretar un dígito en un teclado físico.

Las direcciones IP y el Puerto tienen formatos conocidos; por ejemplo, una IP de 32 bits válida, consta de cuatro conjuntos de números del 0 al 255, cada uno separado por un punto. Una vez que el usuario haya insertado los datos necesarios para realizar la conexión y, antes de intentar efectuarla, se comprueba que dicha información sea válida.

Para ello se emplean expresiones regulares, es decir, una secuencia de caracteres que conforman un patrón de búsqueda, para encontrar una determinada combinación de dichos caracteres, dentro de una cadena de texto.

Si el texto contenido en el campo IP no coincide con el patrón de búsqueda de una IP válida, no se procede a iniciar la conexión y, se muestra al usuario un mensaje de error, avisándole que el formato de la IP introducida no es correcto. Lo mismo ocurre con el campo de texto utilizado para el puerto del servidor.

Una vez verificado que los datos introducidos son sintácticamente correctos, se inicializa la conexión con el servidor. Para ello se crea una conexión UDP cliente – servidor, donde el cliente intenta conectarse al servidor que posee las direcciones indicadas.

Se ha optado por utilizar el protocolo UDP ya que, la conexión, va a ser utilizada principalmente para la transmisión de vídeo en tiempo real. Pese a que el protocolo elegido no verifica la correcta recepción de la información (lo cual puede causar la pérdida puntual de algún fotograma), sí nos asegura una mayor velocidad de transmisión.

El protocolo TCP, por el contrario, sí que posee un sistema de verificación, que asegura la recepción de la información a costa de una menor velocidad de transmisión. Por esta diferencia, el protocolo UDP es el más utilizado por los servicios de transmisión de voz o vídeo en tiempo real, donde la velocidad de la transmisión es más importante que una posible pérdida puntual de datos.

Si el sistema no consigue conectarse al servidor, a pesar de que la dirección sea sintácticamente correcta (bien porque no hay ningún servidor a la escucha en esa dirección o, por otros motivos), se avisa al usuario a través de un mensaje de error.



Adicionalmente cabe destacar que, si en un momento determinado se verificase la pérdida de la conexión, la aplicación mostrará al usuario el motivo de dicha pérdida y volverá a la página de conexión.

Para ello, se recogen las excepciones generadas a nivel de código que contienen la información del fallo y, además de mostrar al usuario un mensaje de fácil comprensión, se muestra la información técnica y detallada para que el personal competente pueda entender qué ha ocasionado el error.

Una vez obtenida la conexión, el sistema muestra la ventana de configuración, desde donde se puede visionar el estado de la línea de producción en tiempo real, y, simultáneamente es posible configurar la cámara.



Figura 5-11 Ventana de configuración del sistema.

\*Imagen en el interior, de la cadena productiva desenfocada por motivos legales.

En la ventana de configuración, se distinguen claramente dos paneles: el primero, de vídeo streaming, que nos muestra las imágenes captadas por la cámara en tiempo real y, donde se diseñan las diferentes áreas de detección que conformarán la configuración del servidor.

El segundo, como podemos apreciar en la imagen anterior, es el llamado panel multifunción, donde se encuentran todos los pulsadores con los cuales el usuario puede interactuar y, realizar distintas acciones.

En la primera conexión, donde aún el usuario no ha realizado una configuración, se muestra como ejemplo una estructura ideal para simplificar su trabajo; estructura que necesita de modificaciones para ser adaptada a las diferentes líneas.

Dicha estructura está compuesta por las diferentes áreas y, cada una de ellas, se indica con un color distinto para que su identificación sea rápida e intuitiva. Como podemos observar, se distinguen las 2 áreas, cuya función es la localización del perímetro, en color amarillo y azul, situadas a los extremos de la imagen, y, las 3 áreas, de detección de movimiento, en la parte central de la misma.

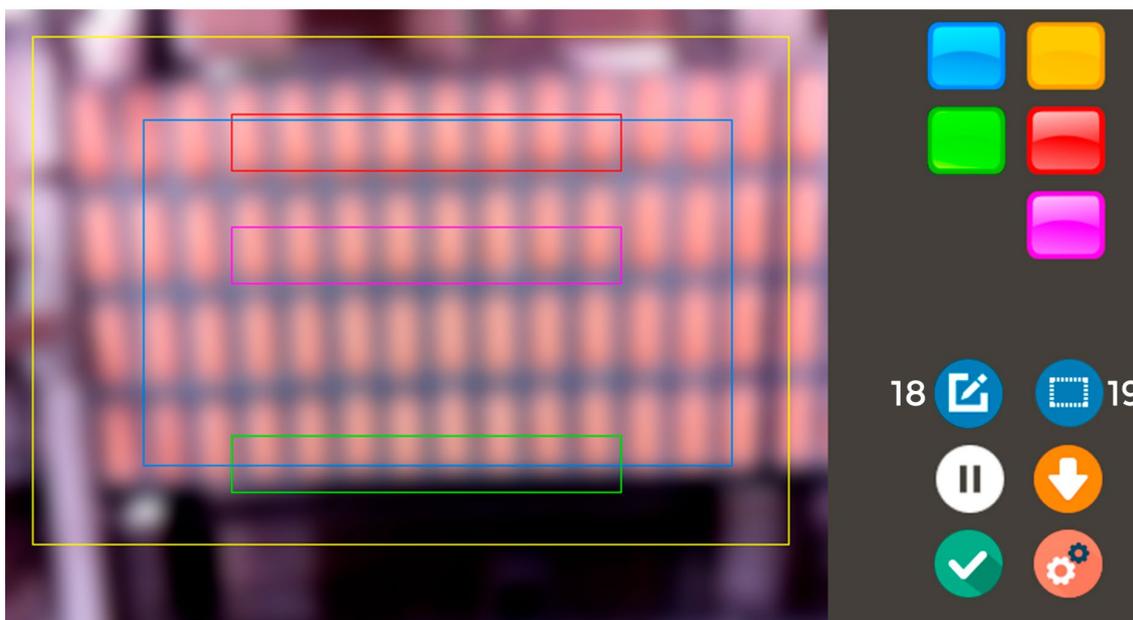
Por medio de los botones de selección, situados en la parte superior del panel multifunción, el usuario puede escoger una determinada área para proceder a su edición. A cada pulsador corresponde un área distinta; éste es fácilmente reconocible ya que el botón asociado a un área dada tiene la misma tonalidad que esta última.

Dentro de la ventana de configuración, se puede trabajar en dos modalidades distintas: la primera es la modalidad que podemos observar en la figura anterior, llamada desplazamiento; el usuario puede saber que está en dicho modo por el botón con el icono de la flecha orientada en las 4 direcciones.

Esto significa que, las áreas podrán desplazarse libremente a lo largo de la imagen, pero sus dimensiones y forma se bloquearán y se mantendrán invariadas; es decir, si el usuario selecciona un área dada, podrá a través del joystick trasladarla donde desee, pero no le será ofrecida la posibilidad de modificar su tamaño.

Sin embargo, si se cambia de modalidad, presionando en el botón de desplazamiento mencionado anteriormente, el panel multifunción varía ligeramente su aspecto y el sistema adopta la segunda modalidad, la de edición.

## Ventana de configuración



### Nuevas acciones disponibles en modo edición:

18 Modo edición.

19 Selección lado área.

*\*Imagen de la cadena productiva desenfocada por motivos legales.*

*Figura 5-12 Ventana de configuración del sistema en modo edición.*

En esta modalidad, el usuario podrá variar las dimensiones de las áreas de detección de perímetro y de movimiento, editando el tamaño de los lados que delimitan dichas áreas (más adelante explicaré su funcionamiento).

Como podemos observar en la figura anterior, el botón que indicaba que nos encontrábamos en la modalidad desplazamiento ha desaparecido, y en su lugar aparece el que nos muestra que estamos en modo edición, y a su derecha encontramos el botón de selección del lado del área.

Dicho botón refleja, qué lado ha sido seleccionado para su posterior modificación; únicamente, en el primer acceso por parte del usuario al modo edición, se muestra que no se ha seleccionado ninguno de los lados. Pulsando dicho botón, se eligen a rotación uno de los 4 lados disponibles; adicionalmente éstos se pueden escoger a través de los botones físicos presentes en el mando de control.

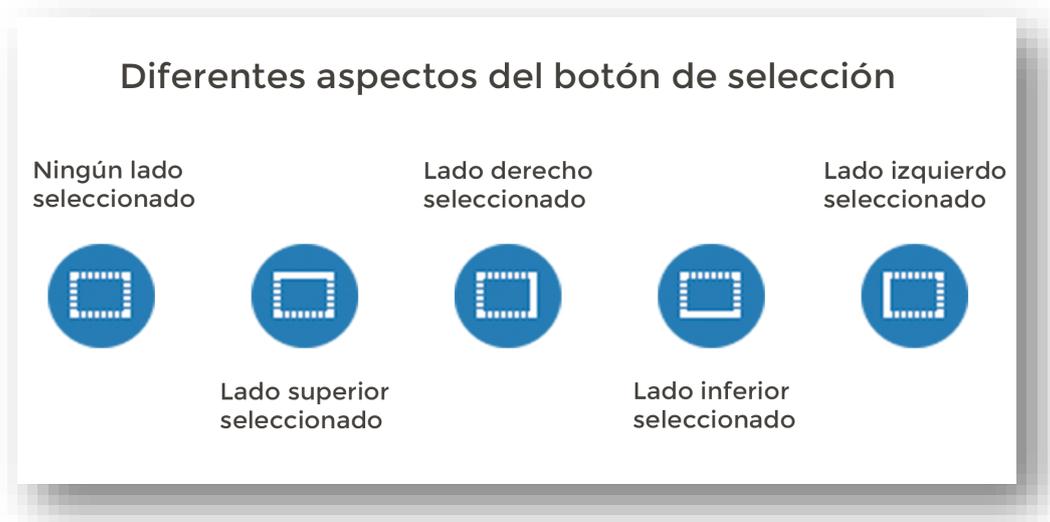


Figura 5-13. Diferentes aspectos del botón de selección de lado.

Siguiendo el análisis del panel multifunción, encontramos el botón para pausar y reanudar el vídeo en directo; esta funcionalidad se ofrece para simplificar y agilizar la creación de una nueva configuración ya que, es mucho más sencillo colocar y editar las diferentes áreas, teniendo una imagen representativa fija en vez de en movimiento.

A su derecha, se encuentra el botón que representa la dirección de la cinta transportadora; ésta puede moverse en dos sentidos distintos, mirando la imagen de arriba hacia abajo y de abajo hacia arriba. Dichos sentidos de marcha están representados a través del icono en forma de flecha presente en el botón.

Es necesario indicar la dirección de la cinta transportadora ya que, se precisa de este dato para calcular correctamente el movimiento en la imagen. Asimismo se ofrece esta acción ya que, dependiendo de la instalación, es posible que sea necesario montar la cámara en sentido contrario respecto a la cinta.

Los últimos dos botones que podemos observar son: el del envío de la configuración al servidor, que si es presionado manda las nuevas disposiciones al servidor, y a su derecha, el de terminar la configuración, que si es pulsado desconecta el cliente del servidor y hace que el sistema vuelva a la pantalla de conexión.

Después de haber ilustrado los componentes que podemos encontrar en la pantalla de configuración, procederé a explicar su funcionamiento y, cómo el usuario, además de, simplemente ver en tiempo real las diferentes líneas de producción, pueda crear nuevas configuraciones y aplicarlas a las diferentes cadenas productivas.

Sobre el panel de imagen, internamente, el software simboliza el eje cartesiano; de esta manera cada área que vemos es definida por una tupla de coordenadas X e Y, que trazan el punto a partir del cual se representan, ella misma, su altura y su longitud.

El usuario, mediante el joystick puede cambiar estos valores, y por lo tanto, el programa se ocupa de reconocer las acciones realizadas por este último, para posteriormente tratarlas y renderizar los cambios.

Para hacerlo posible, el programa está constantemente a la escucha de la información que le llega a través del puerto serie proveniente del arduino; cuando detecta la llegada de una trama pasa a su fase de análisis para reconocer qué acción ha realizado el usuario.

Para identificar los datos contenidos en la trama y saber qué es lo que se desea realizar, se utilizan expresiones regulares. Como he explicado anteriormente en la sección de Aplicación Arduino, las tramas que llevan la información están compuestas por palabras, opcionalmente seguidas por valores numéricos.

Por medio de los patrones generados con las expresiones regulares, se lee cada trama singularmente, ya que éstas son separadas las unas de las otras con un salto de línea. Seguidamente, con la ayuda de un patrón que busca palabras, formadas únicamente por caracteres alfabéticos, se obtiene el primer segmento de la trama.

Una vez obtenido dicho segmento, se analiza para saber qué acción desea realizar el usuario. A continuación, se muestran las correlaciones entre los valores que puede asumir el segmento analizado (según las operaciones realizadas por el usuario), y las acciones que debe ejecutar la aplicación.

Acción usuario	Valor 1º segmento	Acción a realizar
	UP	Pasar a modo edición si ésta no es la modalidad actual y seleccionar lado superior del área.
	BOTTOM	Pasar a modo edición si ésta no es la modalidad actual y seleccionar lado inferior del área.



RIGHT

Pasar a modo edición si ésta no es la modalidad actual y seleccionar lado derecho del área.



LEFT

Pasar a modo edición si ésta no es la modalidad actual, y seleccionar lado izquierdo del área.



X

Desplazamiento o edición del objeto seleccionado, a lo largo del eje X



Y

Desplazamiento o edición del objeto seleccionado, a lo largo del eje Y

---

	SEL	Seleccionar la siguiente área disponible.
---	-----	---

---

*Tabla 9. Correlación acciones usuario en el mando de control – acciones a realizar en el Sistema.*

Cabe destacar que, para ahorrar recursos y tiempo de ejecución (ya que la única acción que el usuario puede realizar sin tener escogida previamente una zona, bien sea de detección de movimiento o de perímetro, es la de seleccionar un área), el software, en estos casos comprueba únicamente que, el valor del primer término sea correspondiente al que se obtiene apretando el botón selección.

Cada vez que éste es pulsado, se escoge un área distinta de las disponibles. Visualmente, dicha acción se observa en la pantalla ya que, el botón correspondiente a dicha superficie desaparece del panel multifunción; es decir, supongamos que al apretar una vez, se selecciona el área de detección de perímetro interna, podremos apreciar que el botón correspondiente, en este caso el azul en forma de rectángulo, desaparece.

Si el usuario vuelve a presionar el botón selección, se escogerá la siguiente área, la de detección del perímetro externo, y observaremos que, el botón amarillo con forma de rectángulo aparece. Así sucederá con todas las áreas; una vez escogidas en su totalidad, a la siguiente actuación se volverá a seleccionar el área inicial.

Sin embargo, si el usuario ya tiene seleccionada un área dada de entre las posibles, la aplicación incluye ya en las comprobaciones los casos restantes. Si la acción corresponde a la de selección de un lado del área, el sistema automáticamente ( si no estamos en modo edición), pasa a éste y selecciona el lado asociado, mostrando al usuario los iconos correspondientes en el panel multifunción.

Escogido un lado, el usuario interaccionando con el joystick puede aumentar su tamaño o desplazarlo, mientras que para disminuir su tamaño es necesario mover uno de los dos lados perpendiculares a éste.

Es decir, supongamos que tenemos seleccionado el lado superior de un rectángulo y que se encuentre paralelo al eje X; si movemos el joystick a lo largo del mismo eje hacia la derecha, el lado elegido aumentará su longitud hacia la misma dirección; y si lo movemos en sentido opuesto, hacia la izquierda, el lado aumentará su longitud en dicha dirección.

Sin embargo, si movemos el joystick a lo largo del eje perpendicular a éste, el Y, el lado se desplazará en el mismo sentido que se ha desplazado el joystick,

incrementando o disminuyendo la longitud de los lados perpendiculares conectados en los extremos al lado que se está editando.

En efecto, si desplazamos el joystick en el eje Y hacia arriba, el lado se moverá en esa dirección y se incrementará la longitud de los dos lados perpendiculares por el mismo valor del desplazamiento. Si se mueve hacia abajo, el lado se trasladará en esa dirección disminuyendo la longitud de los dos lados perpendiculares por el mismo valor del desplazamiento.

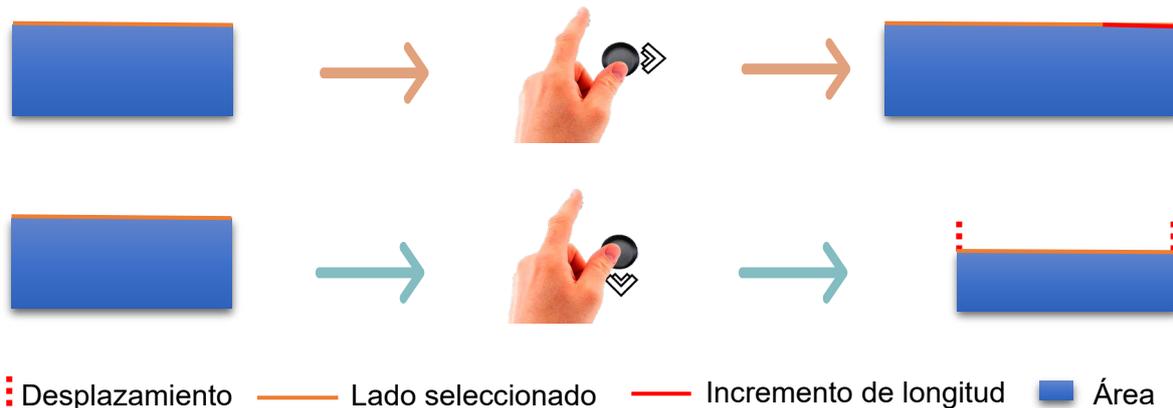


Figura 5-14. Explicación desplazamiento e incremento del lado de un área.

Estas operaciones se pueden realizar sólo en modalidad edición, ya que no es posible mover o editar un lado sin afectar a las dimensiones de los demás. Sin embargo, en modo desplazamiento, el usuario moviendo el joystick en cualquier dirección puede desplazar un área dada, después de haberla seleccionado en ese sentido, sin alterar sus dimensiones.

¿Cómo sabe la aplicación cuál es el valor de dicho desplazamiento o modificación? Si detecta que el primer término de la trama corresponde a uno de los dos ejes, el programa pasa a analizar la segunda parte de ésta y, utilizando expresiones regulares, busca y obtiene los valores numéricos contenidos.

Estos valores, como se ha explicado en la sección de la aplicación Arduino, son las magnitudes del desplazamiento o de la edición. Por ejemplo, si la trama es X-3, el primer término identificará el eje, mientras que -3 corresponde a los píxeles que se debe mover o incrementar el elemento, desde la posición actual, en el eje encontrado anteriormente.

Cada vez que se realiza una operación de movimiento o edición, el software:

1. Obtiene los valores relativos a dicho proceso.
2. Diferencia si se trata de una tarea en modo edición o movimiento
3. Verifica que realizando la operación no se superen los límites de la imagen, es decir, que las áreas tengan una dimensión superior a la imagen o los lados de éstas se representen fuera de ella.
4. Si el proceso no viola los límites de la imagen, renderiza las nuevas coordenadas en la posición correcta, y/o adapta, alto y ancho del área.

Cabe destacar que, no hay ninguna otra restricción además de la mencionada anteriormente; las áreas pueden tener cualquier tamaño, desde ocupar la imagen entera hasta ser diminutas y poseer el tamaño de un píxel; pueden invertirse vertical y horizontalmente.

Todas estas operaciones son necesarias para adaptar la configuración de partida a las varias líneas de producción; el resultado debe ser una estructura donde los bordes de la bandeja se encuentren entre el área de detección del perímetro externo e interno, a una distancia equidistante de éstas lo más pequeña posible.

Las áreas de detección de movimiento deben de ser colocadas respetando el sentido de marcha de la cinta transportadora; ello implica que, la primaria debe de ser la primera por la cual pasará la bandeja con los productos, la secundaria vendrá después, y por último, la terciaria.

La colocación óptima de las áreas de movimiento ve éstas dispuestas una en cada sector de la imagen; es decir, la primaria en la parte inicial, situada sobre el extremo de la bandeja, la secundaria en el sector central y, situada sobre la mitad de la bandeja, y la terciaria en la parte final de la imagen, con el lado del área de detección del perímetro interna que la divide por la mitad.

Ésto se realiza para captar exactamente, cuándo la bandeja con los productos se encuentra centrada en su totalidad en el interior de la imagen, ya que si se dispusieran muy cercanas entre ellas, se detectaría movimiento en todas, pero captaríamos durante su paso sólo una parte de dicho contenedor. Adicionalmente, se recomienda que éstas tengan las mismas dimensiones y, de ningún modo se solapen entre ellas, ya que su efectividad se vería reducida.

Sin embargo la disposición comentada anteriormente permite que, el único instante que se detecte movimiento en las tres zonas simultáneamente sea, cuando la bandeja esté enmarcada en la posición ideal para su análisis posterior e inmediatamente después no se detecte movimiento en el área primaria ya que la bandeja habrá salido de su alcance.

Para poder configurar correctamente la estructura, el usuario espera el paso de una bandeja, y cuando cree que tiene una toma centrada de la misma y de buena calidad, pausa la reproducción del vídeo en streaming. Así, puede colocar tranquilamente los diferentes elementos de la configuración sobre la imagen, y adaptarlos a la línea.

Posteriormente, indica en qué sentido de marcha se está moviendo la cinta transportadora a través del botón de dirección; por último, envía los cambios realizados al servidor, presionando enviar configuración. Una vez que el usuario vuelva a la pantalla de conexión, el servidor comenzará el análisis del vídeo con la nueva configuración indicada.

La configuración que se envía al servidor está contenida dentro de un objeto JSON en formato de texto. Este tipo de objetos, son ampliamente utilizados para el intercambio de información entre diferentes sistemas ya que, JSON es un formato ligero basado en texto plano, donde leerlo y escribirlo es simple para los seres humanos, mientras que, para las máquinas es simple interpretarlo y generarlo.



Otra ventaja que justifica su larga adopción es, que al ser un formato independiente de cualquier lenguaje de programación, los sistemas que comparten información con este método pueden ser desarrollados con diferentes lenguajes.

JSON está constituido por dos tipologías de estructuras: la primera, es una colección de pares nombre - valor, mientras que la segunda, es una lista ordenada de valores; para nuestro objetivo se han utilizado ambas.

La colección de pares nombre- valor, comienza con una llave de apertura y termina con una llave de cierre; dentro, cada nombre es seguido por dos puntos, y los pares nombre - valor están separados por una coma. La lista, comienza con un corchete izquierdo y termina con uno derecho; en su interior, los valores se separan con una coma.

El JSON que contiene la configuración, está formado por una colección de pares nombre - valor que identifica la dirección de la cinta transportadora, y una lista ordenada de valores, llamada "config", dentro de la cual se encuentran tantas colecciones de pares nombre - valor como las áreas presentes. En el interior de estas colecciones se guarda la información de cada área:

- Nombre del área.
- Coordenada X de partida.
- Coordenada Y de partida.
- Anchura de la figura.
- Altura de la figura.

Por último, es muy importante hacer hincapié en el sistema utilizado para representar el vídeo recibido desde el servidor en tiempo real. Este vídeo está formado por fotogramas enviados por la propia cámara a una frecuencia predeterminada.

El cliente es capaz de detectar automáticamente cuándo recibe un nuevo paquete por parte del servidor. Cada paquete contiene un fotograma, que no es más que una imagen, y está codificado en un vector que, a su vez, contiene los valores RGB de cada píxel que la componen.

En el momento que recibe el vector, el programa extrae cada píxel para recomponer el fotograma, hasta mostrar la imagen con sus dimensiones reales en la pantalla del usuario.

## 5.2 SERVIDOR

En esta sección, describiré el funcionamiento del programa que se ejecuta en el servidor, sopesando las características más interesantes. Dicha aplicación se encarga principalmente de analizar el vídeo captado por la cámara y actuar según la información que extrae, mientras que, cuando un cliente se conecta a ella proporciona el vídeo en tiempo real.

También hablaré, a nivel de hardware, sobre el montaje y las interconexiones de los diferentes componentes empleados para construir el servidor. Acto seguido, explicaré brevemente cuál sería su mejor colocación posible, de manera que se obtenga una imagen sin desperfectos.



### 5.2.1 Hardware

El montaje del servidor es mucho más sencillo que el del mando de control, ya que se involucran sólo dos componentes. Dicha sencillez es un requisito para que sea fácilmente escalable a las restantes líneas de producción.

Los componentes necesarios para su realización son: el módulo de la cámara donde está presente el sensor óptico y el objetivo, y la Raspberry Pi 3 B+ que nos permite transformar la cámara en una cámara inteligente, ya que se ocupará de procesar toda la información que es captada por el sensor óptico y atender las peticiones del usuario.

La conexión del módulo con el ordenador se realiza fácilmente a través de un cable plano, que conecta el puerto CSI del dispositivo de vídeo con el de Raspberry Pi. Mediante el mismo podemos obtener las grabaciones e interactuar con la cámara.

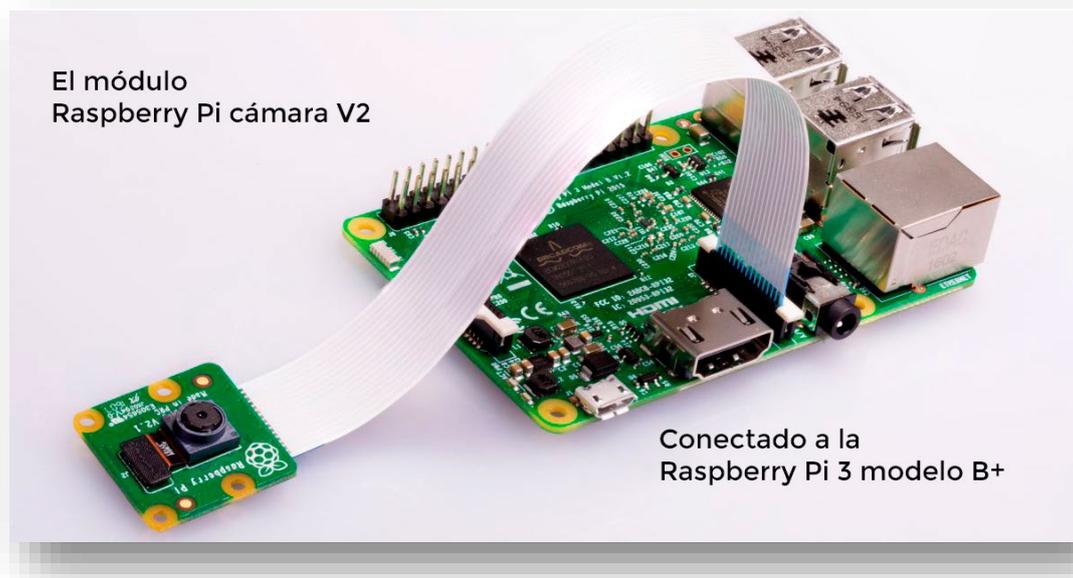


Figura 5-15. Servidor ensamblado. Imagen extraída de [9].

Para un correcto funcionamiento que permita la obtención de una imagen lo más nítida posible, capaz de representar enteramente a las bandejas, se ha instalado el servidor en un soporte colocado paralelamente respecto a la cinta, a 1 m de altura y centrado respecto a la citada cinta.

En esta línea dada se ha determinado con varias pruebas exhaustivas, que dicha posición y altura son ideales para el fin para el cual se va a utilizar la cámara; no obstante, en otras líneas, por limitaciones de espacio u otras problemáticas su colocación puede diferir de la mencionada anteriormente.

Se aconseja, fijar la cámara en una posición elevada respecto a la cinta transportadora, donde la altura respecto a la misma debe ser suficiente para permitir obtener un enfoque claro y completo del ancho de la cadena transportadora y, de buena parte de su sección longitudinal. Simultáneamente, no debe ser muy elevada,

ya que se deben poder distinguir claramente los productos mientras son transportados a lo largo de la línea productiva.

Se obtienen ligeras mejoras, si la cámara se encuentra centrada respecto al ancho de la cinta transportadora; de no ser posible su colocación en dicho punto, se puede colocar en uno de los bordes externos.

La iluminación debe ser lo más uniforme posible; en esta implantación no ha sido necesario ningún dispositivo, ya que la iluminación presente era lo suficientemente buena para poder realizar la solución propuesta.

### **5.2.2 Software**

El servidor se inicializa en un puerto y una IP libres que le son asignadas; esta información se da a conocer al usuario para que pueda conectarse y enviar las peticiones a dicho servidor.

Después de su inicialización, el programa carga una configuración presente por defecto y crea un repositorio en la memoria interna del ordenador donde guardar los eventuales errores; éste tiene en su nomenclatura la fecha y hora de dicha operación.

Tal acción se realiza cada vez que se detecta una configuración nueva, ya que es necesario separar los diferentes resultados que se obtienen con distintas disposiciones y, para poder identificar fácilmente qué estructura se ha utilizado, se incluye la referencia temporal mencionada anteriormente en el nombre del repositorio.

Adicionalmente, dentro de esta carpeta se crea un fichero de texto que contiene todos los registros de las operaciones que ejecuta el programa, con eventuales posibles errores que se puedan producir. Ello se ha concebido para que el técnico informático asignado al proyecto, pueda tener constancia de qué se ha realizado con cada nueva configuración, no sólo visionando personalmente la ejecución sino teniendo un documento que poder consultar en un momento dado.

A continuación, se envía la orden a la cámara de empezar la grabación del vídeo con una resolución determinada y, a partir de este momento el programa empieza a recibir los fotogramas que lo componen.

Si se detecta que hay un cliente conectado al servidor, dichos fotogramas le serán enviados para que pueda controlar la situación en la línea de producción observada. El proceso de envío se realiza de la misma manera cada vez que se detecta un nuevo fotograma; dicha operación es explicada en la siguiente imagen.



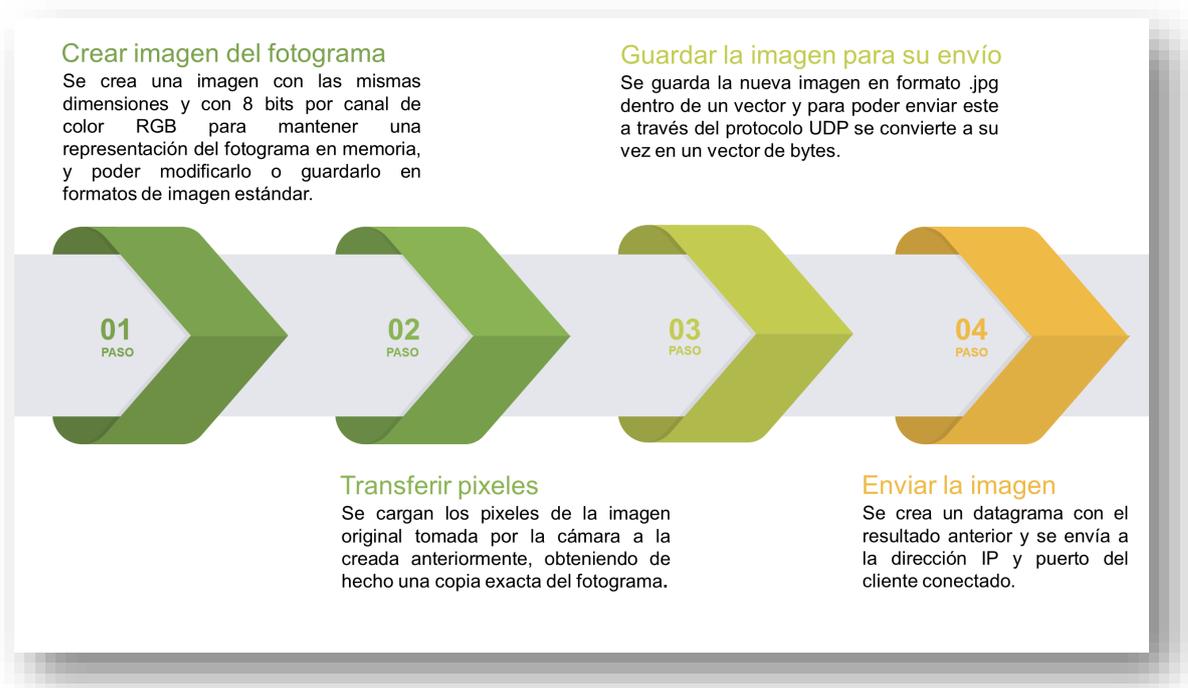


Figura 5-16. Etapas del envío de una imagen desde el servidor al cliente.

Mientras el cliente mantenga la conexión con el servidor, se continuará ejecutando la operación explicada anteriormente para proporcionar a este último el servicio de vídeo en tiempo real.

Si se detecta la recepción de un mensaje enviado por el cliente, significa que ha sido recibida una nueva configuración y se procede a cargarla en el sistema. Para ello se transforma el mensaje en un objeto JSON y se asignan los valores de los parámetros contenidos a sus correspondientes parámetros en el servidor.

Si no se detecta ningún cliente conectado o se interrumpe la conexión existente, el servidor pasa desde la modalidad de configuración a la modalidad operativa y, empieza a analizar el vídeo obtenido por la cámara.

La primera etapa en el proceso de análisis es, individuar si está pasando un objeto de interés (la bandeja con los productos) en el vídeo, y para ello se utilizan las tres áreas de detección de movimiento.

En cada una de ellas, a través de un particular algoritmo el software analiza el flujo de movimiento de cada uno de los píxeles presentes en el interior de estas áreas, comparando los píxeles del fotograma actual con los del siguiente, detectando las diferencias que pueda haber.

A continuación, se realiza el sumatorio de los flujos individuales obtenidos para encontrar el flujo total del área. Dicho resultado se ve influenciado por el movimiento de la cinta transportadora ya que, si ésta es grabada en el vídeo moviéndose desde arriba hacia abajo el flujo es positivo, mientras que en el sentido contrario el flujo resultará negativo.

Para solucionar este problema, se utiliza el parámetro de dirección de movimiento de la cinta, que es proporcionado por el usuario; de esta forma el programa sabe el sentido de la cinta transportadora. Si la dirección es de abajo hacia arriba se calcula el valor absoluto del flujo total para poder realizar las siguientes comprobaciones.

Se verifica si el resultado obtenido es mayor a un umbral que determina si el movimiento que se ha registrado es relevante y por lo tanto significa que, la bandeja está siendo transportada en ese momento. En caso afirmativo se pasa a la siguiente operación, mientras que en el caso contrario se vuelve a empezar el procedimiento calculando un nuevo flujo.

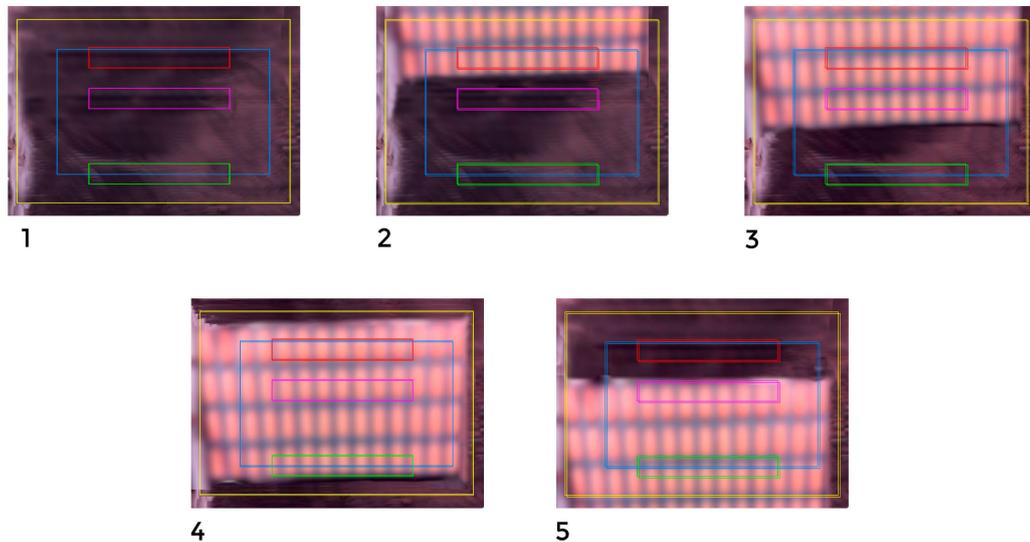
Si se detecta desplazamiento en las tres áreas de detección de movimiento simultáneamente, esto significa que probablemente una bandeja u otro objeto ha pasado por ellas y se encuentra en la posición óptima para ser analizada. Para llegar a esta conclusión, el programa verifica que el flujo total del área primaria de detección sea mayor que el umbral, de la forma explicada anteriormente; si lo es, pasa a calcular el flujo total del área secundaria y vuelve a calcular el del área anterior.

Si ambos flujos tomados individualmente son mayores del umbral, significa que un objeto está pasando por estas dos zonas, por lo tanto, se pasa a calcular el flujo de la tercera área y se recalcula el de las dos anteriores.

Si las tres áreas registran un flujo total de cada zona mayor del umbral, significa que un objeto está atravesando las tres áreas y se encuentra en la posición adecuada para ser analizado. Sin embargo, si una de las comparaciones da resultado negativo, como señalado anteriormente, se vuelve a empezar el proceso de detección calculando el flujo total del área primaria.



## Algoritmo detección de movimiento



### 1 Fotograma de partida con ausencia de movimiento.

Fotograma de la cinta captado por la cámara, donde el flujo de movimiento en las áreas se mantiene por debajo del umbral de detección.

### 2 Movimiento en el área primaria

Se detecta movimiento en el área primaria ya que se asiste a un significativo cambio en los píxeles de la imagen, debido al tránsito de la bandeja.

### 3 Movimiento en el área secundaria.

La bandeja alcanza la segunda área de detección y se verifica el movimiento en la imagen, que causa con su paso. Simultáneamente, se mantiene el movimiento en el área primaria

### 4 Movimiento en el área terciaria.

La bandeja alcanza la última área de detección, se registra movimiento aquí y en las áreas anteriores. La bandeja se encuentra en la posición óptima para la detección del perímetro.

### 5 Ausencia de movimiento en el área primaria.

La bandeja abandona el área primaria. Se asiste a un cambio significativo en los píxeles, no obstante el flujo que se obtiene a continuación es menor al umbral de detección.

*\*Imágenes de los productos desenfocados por motivos legales.*

*Figura 5-17. Ejemplo funcionamiento del algoritmo de detección de movimiento.*

Ahora que sabemos que un objeto está atravesando las tres áreas, se debe comprobar que éste sea el recipiente con los productos que estamos buscando, y para ello pasamos a verificar que el perímetro de dicho objeto corresponda al de la bandeja en cuestión. Es decir, que posea la idéntica forma rectangular y tenga las mismas dimensiones.

El proceso de detección del perímetro se sirve de diferentes algoritmos; el primero que se ejecuta está especializado en hallar los bordes de los elementos presentes en la imagen. Resumiendo brevemente su funcionamiento, el algoritmo compara la luminosidad de grupos de píxeles y establece los bordes, de acuerdo a variaciones en



dicha magnitud, en base a un valor umbral mínimo y otro máximo que le son proporcionados.

Los resultados varían significativamente según con qué umbrales se ejecuta el procedimiento ya que, según qué bordes de determinados elementos se quiere resaltar, trabajan mejor unos valores frente a otros. Si tomamos como ejemplo nuestra bandeja, que es de forma rectangular, tenemos un rango de valores con los cuales obtendremos buenos resultados; dentro de este rango hay umbrales que permiten, según el tipo de producto, que éste no interfiera con la detección del perímetro.

Por lo tanto, es muy importante de cara al éxito del proyecto, elegir los umbrales que mejor resaltan los bordes del objeto de interés, de manera que resulten nítidos, sin desperfectos y claramente visibles. Al mismo tiempo, es conveniente que los bordes de elementos no pertinentes se erosionen lo máximo posible y no sean apreciables.

El segundo algoritmo que se aplica a la imagen, se encarga de encontrar todas las posibles líneas rectas que en ella se puedan apreciar, tomando una colección de puntos que pertenecen a los límites de los elementos presentes, y localizando todas las líneas en las que se encuentren dichos puntos.

El proceso toma como parámetros: un umbral para poder identificar la colección de puntos, la dimensión mínima que debe tener una línea para considerarse como válida y, el máximo espacio que puede haber entre dos segmentos para considerar que éstos en realidad forman parte de una misma línea. En efecto, si esto ocurre, el programa une los segmentos encontrados y los define como uno único.

La dimensión mínima aceptada para una línea es igual a la dimensión del lado más pequeño de la bandeja, que al ser estándar, es igual para todas las líneas. La distancia máxima para considerar dos segmentos como una unidad, depende de la calidad que consigamos al detectar los bordes en la imagen.

Es decir, si conseguimos unos bordes fraccionados o segmentados y no completos, esta distancia deberá ser mayor respecto a la que se utiliza si conseguimos una imagen con bordes completos e íntegros.

Una vez encontradas todas las líneas en la imagen, debemos diferenciar las que efectivamente representan los bordes de la bandeja, de las que pertenecen a otros elementos o corresponden a efectos no deseados de interferencias luminosas. Para ello se utiliza información proporcionada por el mando de control y por las características de las líneas de producción.

Se conoce que el perímetro de la bandeja está contenido dentro de las áreas de detección de perímetro, por lo tanto, cualquier línea que no esté contenida en este sector pertenece a otro elemento, que no es la bandeja, y consecuentemente no debe ser considerada.

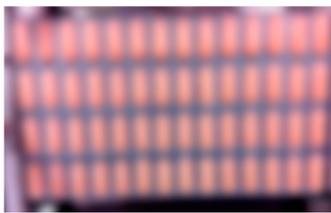
Sabemos que las bandejas transportadas, por el propio movimiento de la cadena o a la hora de pasar de una maquinaria a otra, pueden rotar. El ángulo de rotación propio de las bandejas tiene un máximo y un mínimo conocidos, por ello, si una línea dada posee un ángulo respecto al plano que no se encuentra en este rango, no puede pertenecer al perímetro de la bandeja.



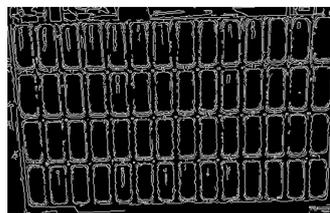
Si unimos todas estas restricciones: tamaño, posición y ángulo, somos capaces de asegurar que una línea pertenece al objeto de interés o, si en el caso contrario pertenece a otro elemento. En efecto, por cada línea encontrada, comprobamos si estos filtros se cumplen y si se encuentran 4 líneas (cada una asociada a un diferente lado del perímetro), identificaremos que el objeto que está transitando en la línea es una bandeja que contiene los productos.

Si no se encuentran las 4 líneas asociadas al perímetro de la bandeja, se deduce que el movimiento detectado ha sido producido por otro objeto o elemento y, se vuelve a comenzar el proceso comprobando si hay movimiento en el área primaria.

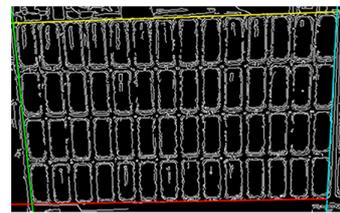
## Algoritmo detección de perímetro



1



2



3

### 1 Imagen resultado de la detección de movimiento.

Al detectarse movimiento en las tres áreas se captura el fotograma.

### 2 Imagen resultado de la detección de los contornos.

Se convierte la imagen en blanco y negro. A continuación, se extrapolan los contornos de los elementos comparando la luminosidad de los píxeles.

### 3 Imagen resultado de la detección del perímetro.

Se buscan en la imagen líneas rectas que respeten las restricciones y representen el perímetro de la bandeja.

*\*Imagen del producto desenfocada por motivos legales.*

*Figura 5-18. Ejemplo funcionamiento algoritmo detección de perímetro.*

Ahora que se conoce que efectivamente estamos en presencia de los productos, se debe poder reconocerlos y cuantificarlos. Para ello se utiliza un concepto basilar en la visión por computador ya explicado anteriormente, es decir, saber distinguir entre los píxeles de interés y los de no interés.

Los productos tienen colores y tonalidades totalmente diferentes del recipiente que los transporta; si se detectan y agrupan estos colores se pueden reconocer los productos y saber cuántos hay.

Para lograrlo se procede a recortar el fotograma, obteniendo como resultado una imagen donde se encuentra sólo la bandeja con los productos. Ello se realiza para evitar englobar en el proceso de detección, objetos presentes en la escena que tengan el mismo color que los productos.

A continuación, se establece como modelo de color el HSB; este modelo es una transformación no lineal del espacio de color RGB donde los colores se representan en términos de sus componentes, que son: Matiz, Saturación y Brillo.

El modelo HSB se basa en la percepción humana del color, y como ya mencionado, al igual que el ojo humano describe tres características fundamentales del color:

- Matiz: “Color reflejado o transmitido a través de un objeto. Se mide como una posición en la rueda de colores estándar y se expresa en grados, entre 0° y 360°.”[\[29\]](#)
- Saturación:  
Fuerza o pureza del color. Representa la cantidad de gris que existe en proporción al tono y se mide como porcentaje comprendido entre 0 (gris) y 100 (saturación completa).[\[29\]](#)
- Brillo: “Luminosidad u oscuridad relativa del color. Se suele medir como un porcentaje comprendido entre 0 (negro) y 100 (blanco).”[\[29\]](#)

Se utiliza este modelo, ya que nos permite mantener tonos específicos al cambiar o limitar el espacio de color. Asimismo, al estar inspirado en el sistema de visión humano es más intuitivo respecto al modelo RGB, especialmente a la hora de crear un tono específico, pues facilita la comprensión de las relaciones que existen entre los distintos colores.

Posteriormente, se aplica a la imagen un filtro transformándola en blanco y negro; durante el proceso se clona la gama de colores original encima del efecto. El copiado de las tonalidades es necesario para realizar sucesivamente el análisis por color, mientras que el filtro se aplica para poder detectar los contornos de dichas tonalidades con más facilidad, gracias a un fuerte contraste entre píxeles de interés y de no interés.

Como expresado anteriormente, los productos tienen diferentes colores y de cada color hay varias tonalidades, por lo tanto y para obtener mejores resultados a la hora de encontrar un determinado color, no se buscará el matiz exacto sino que se creará un rango, con unas magnitudes mínimas y máximas, con variaciones próximas al tono que se desea identificar.

Una vez ha tenido lugar la búsqueda se erosionan los límites de los objetos en primer plano, obteniendo una representación de los colores encontrados más definida y acentuando sus características. Identificados los colores, se obtienen los contornos de sus posiciones, es decir, si en un sector de la imagen se ha encontrado una zona que corresponde a ese color, de esa área se identifica la posición de su perímetro y, posteriormente se guarda en una matriz.

El siguiente paso es, reducir los varios colores identificados a uno sólo; a partir de la matriz se extraen los contornos encontrados anteriormente y se obtiene de ellos una forma geométrica rectangular, aproximada a la superficie que ocupaban.



Si el rectángulo obtenido posee unas dimensiones mayores que un determinado umbral, es decir, el color ocupaba en la imagen una superficie considerable y por lo tanto no es despreciable, se considera como tal. Sin embargo, si la forma geométrica obtenida tiene unas dimensiones muy pequeñas no se considera aceptable y se desprecia, pasando a analizar el siguiente rectángulo obtenido.

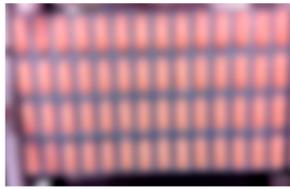
Este proceso se repite para todos los contornos que se encuentran en la matriz y, todos los que son considerados como válidos serán representados sobre la imagen en sus correspondientes posiciones, asociándoles un único color primario. Ahora, en lugar de tener muchas superficies de interés con colores distintos, tendremos el mismo número de superficies con un único color.

El último paso es una repetición de los algoritmos anteriores; se busca en la imagen el matiz del color primario, esta vez sin crear ninguna escala ya que no hay tonalidades; una vez identificado, se guardan las posiciones de sus límites y se consigue la figura geométrica aproximada.

Con este procedimiento la figura que se obtiene es la agrupación de todas las figuras contenidas anteriormente que tenían tonalidades diferentes entre ellas y, representa al producto en su totalidad.

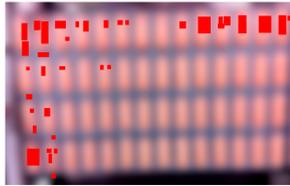


## Algoritmo detección de producto



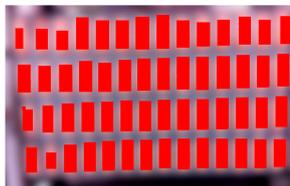
### 1 Recorte de la imagen.

Se recorta el fotograma para que represente solamente la bandeja. De esta manera se evita analizar sectores de la imagen que no son de interés.



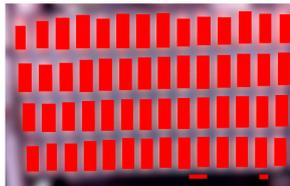
### 2 Búsqueda y representación 1<sup>er</sup> color.

Se busca en la imagen el primer color, se obtiene la representación geométrica de la superficie que ocupa y se le asigna un color primario.



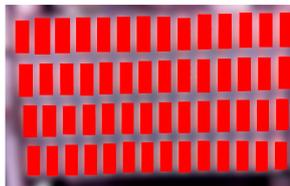
### 3 Búsqueda y representación 2<sup>o</sup> color.

Se repite el proceso de búsqueda con otro color, en este caso el color n<sup>o</sup>2,



### 4 Búsqueda y representación 3<sup>er</sup> color.

Se repite el proceso anterior buscando en la figura el color n<sup>o</sup> 3.



### 5 Sumatorio de los resultados anteriores y detección producto.

Se recopilan todos los resultados obtenidos anteriormente, y las áreas con el color primario en contacto con otras se unen en una única superficie. A continuación, se analizan las dimensiones de las formas geométricas obtenidas y se conoce si representan un producto válido.

*\*Imágenes de los productos desenfocadas por motivos legales.*

*Figura 5-19 Ejemplo funcionamiento del algoritmo de detección de producto.*

Obtenidas las representaciones de los productos, se debe comprobar si estos poseen las dimensiones correctas para su posterior venta, ya que puede suceder que los moldes no se rellenen enteramente. De cada producto se conocen las dimensiones mínimas y por tanto, se verifica que las figuras geométricas superen dichas magnitudes.

Si lo consiguen, se categoriza la representación y el producto como válido, en caso contrario se descarta. Analizadas todas las figuras se realiza un sumatorio de los productos útiles, si el total encontrado es mayor o igual que el límite mínimo de productos que debe ser transportado en una bandeja, significará que no se ha encontrado ningún error en la producción.

Por el contrario, si este número es menor que el umbral dado, significa que hay demasiados moldes vacíos o incompletos en las bandejas y consecuentemente la producción está disminuyendo, bien por un error puntual o bien porque se ha producido un error al origen de la cadena productiva.

Si se da esta situación, el programa almacena en la memoria interna del servidor una foto, sin alteraciones, de la bandeja detectada como errónea y en su nomenclatura se incluye, la fecha y hora de su captura y el número de productos detectados.

Al acabar el estudio de integración de esta nueva aplicación con el sistema de avisos existente, se encontrará el destino óptimo de envío de esta imagen como señal de error, para que el operario, a través del mando de control pueda conectarse a la cámara y ver lo que está sucediendo.

### **5.3 SOFTWARE DE APOYO**

En esta sección se explican dos programas de apoyo a la aplicación principal, que van a ser utilizados no por el usuario final sino por el técnico informático a cargo del proyecto. Sus principales funciones engloban la prueba en tiempo real de los algoritmos utilizados por el servidor, sin tener que instalar el sistema en una cadena de producción operativa.

El primer programa ayuda a determinar los umbrales necesarios para el correcto funcionamiento del algoritmo de detección del perímetro. En lo específico, es de utilidad para verificar si se obtienen los contornos de los elementos en una imagen, si se detectan suficientes líneas para reconocer la bandeja y, cómo es el rendimiento de los valores escogidos en diferentes tipologías de productos.

El segundo programa se ha ideado para conocer qué colores componen un determinado producto, ya que esta información es de vital importancia para un correcto funcionamiento del sistema.

A través de esta aplicación, no sólo podemos conocer estos datos sino que también podemos ejecutar el algoritmo de detección del producto y ver los resultados en tiempo real, obteniendo una retroalimentación inmediata de si los valores elegidos funcionan correctamente.

Ambos programas se han realizado mediante el mismo entorno de programación Processing, y son compatibles tanto para ordenadores convencionales como de placa reducida con variantes de los sistemas operativos Linux o Windows.



### 5.3.1 Configuración algoritmo detección de perímetro

El programa emplea el mismo algoritmo utilizado en el servidor, el usuario puede cambiar los umbrales utilizados para hallar los bordes de los elementos presentes y, adicionalmente modificar los parámetros de entrada para la búsqueda de las líneas (mencionada en la explicación del software del servidor).

Después de haber introducido los parámetros que se desea probar, se pueden cargar hasta 4 imágenes distintas, representativas de un producto único o de varios productos; se aconseja que éstas sean imágenes tomadas desde la misma cámara empleada por el servidor o, capturadas desde orientaciones similares para realizar test lo más precisos y certeros posible.

Después de haber seleccionado las imágenes, el usuario puede modificar la dimensión y la posición de las áreas de detección de perímetro, para que el perímetro de la bandeja quede comprendido entre éstas.

Por último, puede indicar si desea ver todas las posibles líneas asociadas a los lados de la bandeja que es capaz de detectar el algoritmo o, si solamente desea ver una, como ocurre en el software empleado en el servidor.

A continuación, el algoritmo se ejecuta individualmente en cada una de las imágenes que el usuario ha cargado en memoria, y una vez completado el proceso, muestra todos los resultados en pantalla junto a las áreas de detección de perímetro.

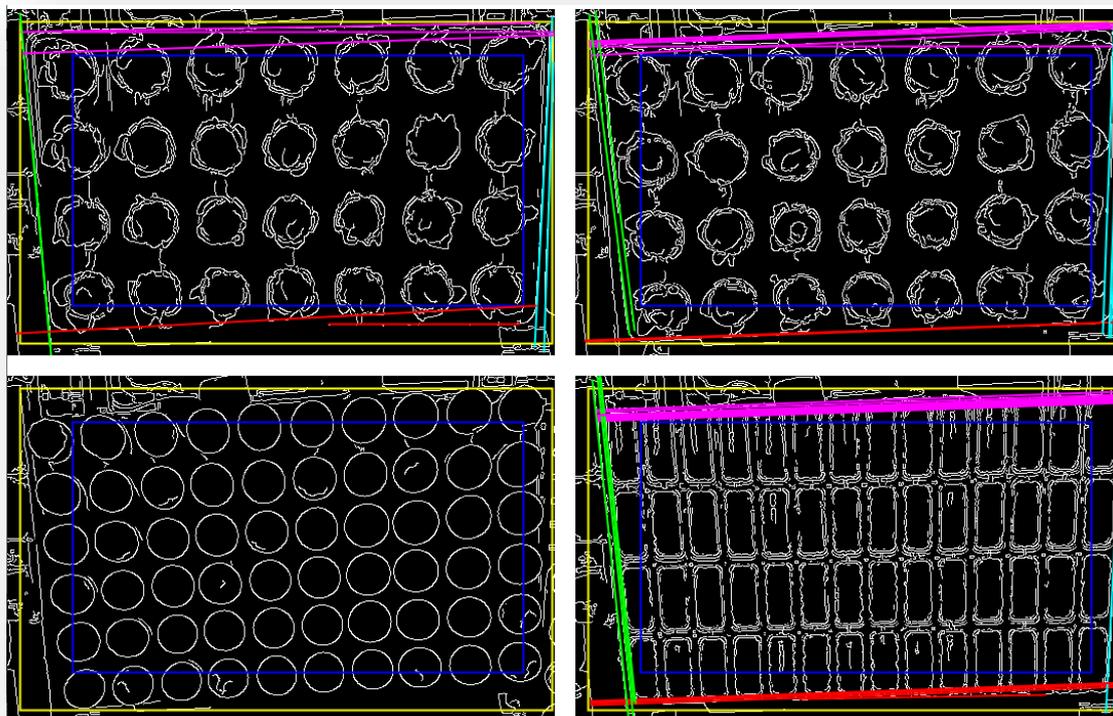


Figura 5-20. Resultados aplicación del algoritmo de detección de perímetro sobre diferentes productos.

Como se puede observar en los resultados de la ejecución que se muestran en la figura anterior, los parámetros insertados trabajan bien con tipologías de productos rectangulares. Se llega a esta conclusión ya que, en la última figura representativa de dichos productos, se han encontrado varias líneas que representan cada lado del perímetro de la bandeja y, los contornos obtenidos de los elementos incluidos en el recipiente no interfieren con la detección de sus límites.

Si se aproxima la forma geométrica del producto a una circular, se puede comprobar que el algoritmo con la configuración actual tiene dificultades para distinguir el perímetro de la bandeja. Se puede comprobar dicha afirmación en las imágenes nº1 y nº2; en la nº1 sólo 3 de los 4 lados se detectan correctamente mientras que en la 2ª sí se obtiene la detección completa del perímetro, pero al igual que en la anterior el número de líneas encontradas disminuye considerablemente.

En la imagen nº3 donde el producto tiene una forma completamente circular, con los parámetros utilizados no se consiguen obtener unos contornos claros y concisos de la bandeja, por lo tanto, el sistema no puede encontrar las líneas asociadas al perímetro.

En resumen, gracias a este programa se puede probar una misma configuración para varios productos, ver en qué tipologías obtiene buenos resultados y conocer los valores óptimos de cada producto. Se obtienen así ahorro de tiempo y de inversión de dinero, ya que no es necesario implementar el sistema en ninguna línea de producción, dicha acción se realizará solamente cuando se haya comprobado el correcto funcionamiento del sistema

### **5.3.2 Configuración algoritmo detección de producto**

Otro de los pilares sobre el cual se basa el proyecto es, la capacidad de detectar un producto dado por su color, pero ¿cómo podemos saber qué tonalidades componen un producto, cuáles son sus correspondientes valores en el modelo HSB y, sobre todo, conocer si las magnitudes elegidas permiten una clara detección del producto?

Este programa se ha realizado para ofrecer una solución a estas problemáticas; en la ventana de inicio se encuentra un menú desplegable que mostrará todas las tipologías de productos, ordenadas por nombre. Por ahora se han configurado y probado dos tipologías por lo que sólo se permite dicha elección.

Una vez se selecciona el producto deseado, se muestra otra ventana con una imagen representativa de la bandeja que transporta dicho producto. El usuario tiene a su disposición como máximo 4 sectores, donde puede almacenar un color en cada uno.

Para guardar un color, se debe presionar la tecla correspondiente al número de dicho espacio y mantenerla pulsada; seguidamente, con el ratón pulsar en la imagen, en el lugar a partir del cual se desea extraer el color. El programa reconoce las coordenadas del lugar donde el usuario ha pulsado, extrae el píxel que se ubica en esa posición y obtiene su color.

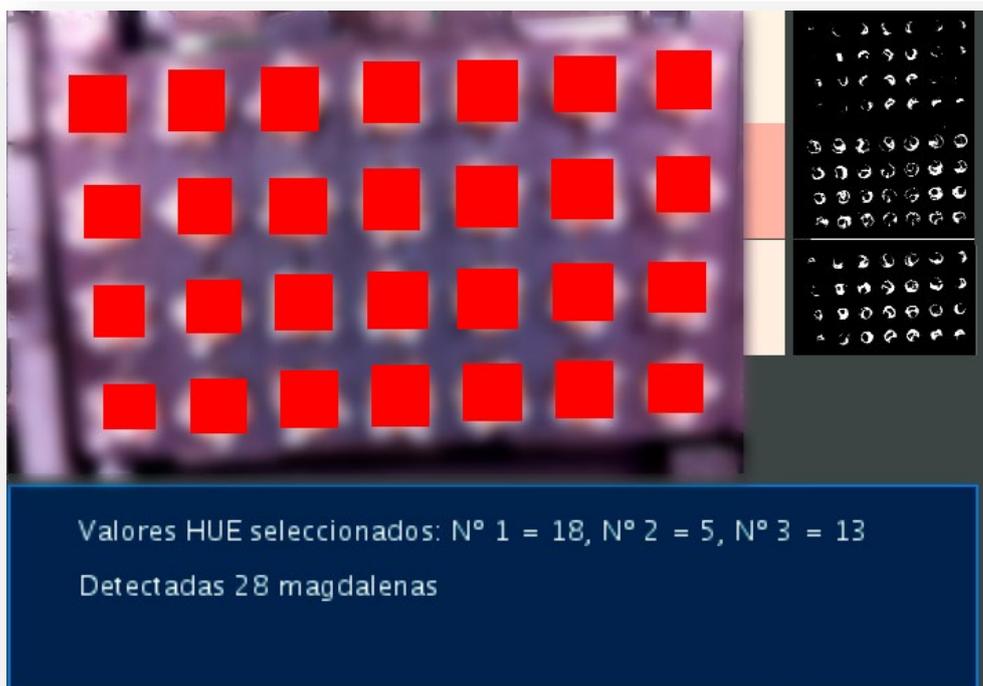


Seguidamente, ejecuta el mismo algoritmo de detección de producto presente en el servidor y, muestra en la imagen qué zonas se han capturado con ese valor mostrando las aproximaciones geométricas correspondientes.

Adicionalmente, aparece en pantalla un recuadro donde se muestra el color elegido, y una imagen en blanco y negro donde se representa detalladamente qué píxeles se han escogido a través de esa tonalidad. Dicha imagen es el resultado de un paso intermedio del algoritmo de detección de producto.

Si el usuario desea cambiar el color guardado en un espacio dado, es suficiente volver a pulsar el mismo número. Cuando se ha conseguido una buena representación de los productos, pulsando la tecla “envío”, se puede activar el algoritmo de detección de producto completo, que a partir de todos los colores elegidos intenta reconocer y cuantificar los productos presentes comprobando siempre si es válido según sus dimensiones.

Cuando el proceso finaliza, se muestra al usuario en la imagen el resultado final y, aparece un mensaje avisándole sobre el número de productos encontrados y qué matiz tienen los colores elegidos en el modelo HSB.



*\*Imagen del producto desenfocada por motivos legales.*

*Figura 5-21. Ejemplo de resultados de ejecución software de apoyo para la detección de los productos.*

---

---

## CAPÍTULO 6

### PRUEBAS

---

Actualmente, el sistema se encuentra implantado en una línea productiva en la cual se fabrican tres tipologías distintas de productos; en esta instalación es donde se verifican mayormente los problemas que se han comentado en los primeros capítulos. Adicionalmente, se ha optado por los diferentes productos que circulan, ya que una mayor variedad permite realizar pruebas diferenciadas y certeras.

La aplicación que se ejecuta en el servidor no incluye el algoritmo de detección de productos, ya que se ha implementado en un segundo momento y, adicionalmente las pruebas que se están ejecutando tienen como finalidad el comprobar la correcta detección de movimiento y la posterior captura del fotograma representativo del contenedor con los productos.

Dichas pruebas no se pueden realizar de otra manera, pues se necesita disponer de la cinta transportadora en sus condiciones operativas. Utilizando las aplicaciones de apoyo, se ha comprobado la eficacia del algoritmo de detección del perímetro, obteniendo los umbrales para la búsqueda de los contornos de dos tipologías de productos.

Además, se han obtenido los colores empleados para la detección de todas las tipologías de productos que, son realizados en la cadena de producción elegida para las pruebas. Se ha verificado su funcionamiento en laboratorio a través del programa de apoyo para la detección de productos obteniendo un resultado satisfactorio; en efecto, el software es capaz de cuantificar la totalidad de los elementos presentes.

En la línea, las pruebas de detección de movimiento evidencian que el correcto funcionamiento del sistema se ve afectado en gran parte por la disposición de las áreas de movimiento, ya que una incorrecta disposición no da la posibilidad de obtener una imagen representativa de los elementos de interés.

De momento, la disposición con la cual se han obtenido los mejores resultados ha sido la que se ha explicado en la sección “Software del servidor”; no obstante, en un futuro ésta podría variar si se encontrasen soluciones más efectivas.

Se ha demostrado la eficacia del algoritmo empleado para la detección de movimiento dentro de las áreas, no obstante, debido a las diferencias entre el ambiente de producción y el de laboratorio (principalmente al trabajar con una imagen dinámica frente a una estática), el algoritmo de detección de perímetro a veces no es capaz de individuar correctamente la bandeja, bien porque no se delimitan los contornos o bien porque no se encuentran líneas asociables al perímetro.

Actualmente, en las pruebas preliminares se ha obtenido una tasa de error más elevada de lo esperado, aproximadamente con una media del 18%. Para disminuir este porcentaje se necesitan más pruebas en la línea variando los parámetros de



entrada para los algoritmos. Asimismo, se han resaltado algunas de las causas que no permiten la correcta detección; a continuación, se describen brevemente cuáles son estas problemáticas, cómo se han abordado y qué soluciones se han encontrado.

El ritmo de producción a veces es irregular, lo cual ocasiona que las bandejas no tengan el tiempo necesario para abandonar las áreas de detección de movimiento sin que éstas se activasen, ya que en las últimas dos se registraba el desplazamiento de salida, mientras que en la primera se registraba el desplazamiento ocasionado por la llegada de un nuevo recipiente.

De esta manera se obtenía un fotograma que representaba dos bandejas parciales. Para solucionar este problema, después que el software realiza una detección de producto se ha introducido un tiempo de espera, durante el cual no se analiza el flujo de movimiento, dando de esta manera tiempo a los recipientes detectados para abandonar las áreas.

Igualmente, se ha planteado mover la localización de la cámara a otra zona de la cinta de producción, donde la frecuencia de paso es más estable y, la velocidad de paso de los productos es menor. Actuando así, se obtiene un mayor control sobre los flujos de movimiento y al mismo tiempo una mayor calidad de los fotogramas capturados.

En un principio, se habían planteado solamente dos áreas de detección de movimiento y una sola de detección de perímetro, con el resultado que: era más fácil que se verificasen falsos positivos, que el fotograma obtenido no estuviese centrado y, que se encontrasen demasiadas líneas asociables al perímetro de la bandeja.

La solución ha sido el sistema actual, es decir, encapsular el perímetro de la bandeja entre dos áreas para tener un área de búsqueda más reducida y, poder distinguirla visualmente e, introducir una tercera área de movimiento para poder tener un mayor control sobre la posición de la imagen, limitando los falsos positivos.

En un primer instante, la detección de producto debía ser realizada a través de otro programa y este sistema debía servir solamente para monitorizar la línea, detectar el paso de las bandejas y, proporcionar imágenes de entrada para la otra aplicación lo más claras posibles

La idea de utilizar la gran diferencia entre los colores del producto y del recipiente, ha nacido observando un pequeño ejemplo de una aplicación, donde el usuario indicaba una tonalidad seleccionándola de una paleta de colores y, a continuación se evidenciaba en la imagen ese matiz. De aquí, se ha extrapolado el concepto para poder detectar los productos y realizar el software de apoyo.

Se ha constatado que, en el sistema, si la producción estuviese parada por error y no transitaran bandejas, no se podría avisar de ello. Para resolver este problema se ha planteado introducir un temporizador, que equivale al tiempo en el cual deberían pasar dos bandejas; el temporizador se reinicia cada vez que se hace un reconocimiento de los productos, sin embargo, si llega a cero quiere decir que se ha verificado un paro de la producción y, por lo tanto se captura la imagen correspondiente y se avisa de ello.

En la primera versión del software del microcontrolador, se enviaban siempre las mismas magnitudes, independientemente de cuanto se desplazase el joystick, incluido el caso de que se moviese por error debido a algún contacto involuntario. Observando



que ello no permitía un manejo fluido del desplazamiento de las áreas, se ha optado por adoptar el sistema de sectores con diferentes valores de movimiento, aumentando así la usabilidad por parte del usuario.

---

---

## CAPÍTULO 7

### CONCLUSIONES

---

Ser puesto al cargo de la realización de un proyecto extremadamente actual y práctico, con una envergadura importante a la vez que, con escaso tiempo para su realización y muchos aspectos diferentes a considerar, desde su planificación hasta su implementación, para una de las empresas más importantes en España como es Dulcesol, ha sido al mismo tiempo una gran satisfacción a nivel personal y un importante reto técnico.

El proyecto realizado ha sido un desafío complejo, debido a las características explicadas anteriormente, que ha dado un significado extremadamente práctico a la definición “Ingeniería del software” a través de todos los pasos ejecutados, desde la detección de los requisitos y su análisis, hasta el desarrollo y su posterior implementación.

Lo que se ha comprendido es que, es necesario enfocar el problema desde diferentes puntos de vista, obtener la mayor información posible, analizarla escrupulosamente y, mantener siempre clara la meta, avanzando poco a poco, pero de manera decidida por diferentes caminos hasta alcanzarla; obteniendo un programa que sea sobre todo útil, funcional e intuitivo, para el usuario.

Comenzando el proyecto las únicas certezas a mi disposición eran: la explicación de un problema existente, numerosos recursos teóricos, una idea de partida, un algoritmo de ejemplo sobre la detección de movimiento y una dirección a seguir (proporcionados por mi tutor del proyecto). El desarrollo se ha realizado implementando las metodologías ágiles, planteándose objetivos y reuniones semanales, para poner en común ideas, resolver problemas, saber dónde nos encontrábamos y hacia dónde queríamos ir.

Alcanzar los objetivos propuestos para la realización del trabajo, obtener algo tangible y reconocible, que pueda ser utilizado por diferentes personas, dejando una traza de mi estancia en la empresa ha sido una nota de alegría y satisfacción en terminar algo que sentía mío, y a la hora de dejar el proyecto, sentir un poco de pena por los pasos que aún quedan por hacer.

En primer lugar, se ha conseguido una herramienta con la cual automatizar la tarea repetitiva analizada, aligerando la carga de los operarios y, de bajo coste respecto a las que están presentes en el mercado. Sobre este aspecto debería hacer reflexionar cómo dos tecnologías, Raspberry PI y Arduino, creadas por la misma motivación,



fomentar la instrucción y acercar las personas al mundo de la electrónica y de la informática, sean hoy en día utilizadas ampliamente no sólo para esta finalidad, sino como elementos de la industria 4.0.

Resaltando cómo la tecnología e investigación, y la instrucción, se influyen las unas con las otras (pues potenciándolas se obtienen repercusiones positivas entre ellas), es aquí, tanto en el presente como en el futuro, donde se deberían dirigir nuestros esfuerzos como sociedad.

En segundo lugar, se ha obtenido un sistema que es escalable a todas las líneas de producción deseadas, con un coste de implementación reducido, ya que la componente que se instala es la que tiene menor coste respecto a las que forman el sistema y, requiere sólo de un soporte fácilmente realizable y muy económico.

Por último, se ha concebido una herramienta totalmente portátil y manejable desde cualquier lugar de la empresa, que necesita solamente de una conexión a internet y, que permite observar en tiempo real desde remoto, el estado de la línea de producción y al mismo tiempo configurar cada software de reconocimiento instalado.

Durante el desarrollo del proyecto se han aprendido tecnologías nuevas que no habían sido abordadas anteriormente, comenzando por los entornos de programación Processing y Arduino, obteniendo un conocimiento que permite su fácil empleo.

Al mismo tiempo se ha adquirido la habilidad para realizar programas destinados a microcontroladores, potenciando el lenguaje de programación C. Se ha investigado sobre la tecnología OpenCV y OpenGL, aprendiendo sus fundamentos y conociendo cómo trabajan los métodos empleados. Se han abordado las principales técnicas utilizadas en visión por computador, obteniendo los conocimientos necesarios para poder aplicarlas a una situación real, utilizándolas entre otras como la solución a un problema detectado.

En el sector hardware, se han alcanzado conocimientos sobre la construcción y el funcionamiento de los ordenadores de placa reducida y de los microcontroladores. Se ha aprendido a trabajar con ellos, a obtener y enviar información a los periféricos, conectados a través de los GPIO. Asimismo, se han obtenido conocimientos básicos del mundo de la electrónica.

En definitiva, este proyecto ha significado el ver aplicados los conocimientos de los años pasados estudiando el grado; un gran reto técnico, tomado con seriedad y, llevado a cabo gracias a la pasión por la informática, la curiosidad por las nuevas tecnologías y la satisfacción de ver los esfuerzos realizados de una manera tangible.



## 7.1 RELACIÓN DEL TRABAJO DESARROLLADO CON LOS ESTUDIOS CURSADOS

El proyecto realizado ha englobado de manera práctica en su desarrollo, numerosos conocimientos y varias técnicas estudiadas en estos años. Posee una gran relación con los conocimientos recibidos en la asignatura de “Sistemas robotizados”, donde se han estudiado las bases y los conceptos de visión por computador, necesarios para entender el proyecto y que han sido profundizados con su desarrollo.

La fase de planificación del sistema se ha realizado, siguiendo las pautas que se han estudiado en numerosas asignaturas, entre ellas “Análisis y especificación de requisitos” y “Proceso de software”, donde se ilustran los conceptos de metodologías ágiles, la importancia de cara al éxito del proyecto de realizar una planificación y, un análisis exhaustivo de la información poseída.

Durante el desarrollo se han empleado conocimientos obtenidos en diferentes asignaturas de programación, entre los más importantes se cita la necesidad de realizar un software de calidad y, el uso de convenciones y patrones sobre la escritura del código, con la finalidad de obtener un código claro, conciso y de fácil interpretación, además de optimizar los recursos del ordenador disponibles. Adicionalmente se han utilizado lenguajes de programación aprendidos y profundizados en dichas materias.

La asignatura de “Redes de computadores” ha sido de ayuda en la planificación y creación de la arquitectura cliente – servidor, de las conexiones entre los varios componentes y, en la elección del protocolo a utilizar para el envío y recepción de la información (ya que todos estos conceptos se han estudiado durante el curso).

En el ámbito de la electrónica se han adoptado nociones aprendidas durante las varias asignaturas que englobaban esta materia, como por ejemplo “Tecnología de computadores”. Dichos conceptos han sido de gran utilidad a la hora de diseñar la estructura hardware del proyecto y conectar los diferentes elementos que lo componen.

En general, se han utilizado conceptos de la mayoría de las asignaturas vistas en la carrera; en particular, se han empleado nociones de casi la totalidad de las asignaturas de la rama de “Ingeniería del Software”. Se han aprendido y puesto en práctica nuevas nociones y tecnologías, potenciando los conocimientos adquiridos durante los estudios.

En conclusión, y como expresado anteriormente se puede afirmar que el desarrollo de esta aplicación, realizando todas las etapas de la vida de un proyecto informático, engloba y sintetiza de manera extremadamente tangible y práctica la definición de “Ingeniería informática”, donde basándome en la experiencia realizada, el ingeniero debe estar siempre dispuesto a reinventarse, a realizar un aprendizaje continuo, poseer una mentalidad flexible, estar abierto al cambio, aceptar el fracaso de sus soluciones pero no obstante, seguir siempre ¡ADELANTE!.



## AGRADECIMIENTOS

---

Quiero empezar agradeciendo en primer lugar, a todos mis compañeros del departamento de informática e I+D y a mi tutor de prácticas en Dulcesol, por hacer que ir a trabajar fuera un placer gracias al buen ambiente y la amistad instaurada.

Una mención especial de agradecimiento va a mi tutor de proyecto, Rubén Palmer, por confiar en mí, ver que tenía potencial y asignarme el trabajo. Has estado siempre a mi lado, enseñándome mucho sobre el mundo de la electrónica y empujándome en los momentos más oscuros, cuando había más errores de compilación que líneas de código.

Gracias por confiarme tu idea, ya que este proyecto es de los dos, por todos los consejos que me has dado y por realizar esa pasada de carcasa en impresión 3D para el mando de control. Todo ha sido posible gracias a ti.

Gracias a mi familia, por aguantarme y animarme durante la redacción del TFG, por ayudarme en la corrección y en el análisis de este último, leyéndolo una y otra vez, como si fuera la primera y por tener esa infinita paciencia.

En especial, muchas gracias a mi tío, por estar siempre a mi lado y ser mi estrella polar aquí en España, a mi madre, por impulsarme en todo momento a estudiar en la UPV y darme esa idea, que se ha convertido en realidad.

Gracias a mi novia, por insistir y darme la idea de realizar las prácticas en Dulcesol, cuando yo no lo creía viable, tenerme cerca cuando estaba insoportable y ayudarme siempre, incluso cuando no tenía el tiempo material para hacerlo; a su familia, por acogerme en su casa durante este tiempo de difícil cuarentena, aconsejarme y hacerme sentir como uno más y cuidar de mi en todo momento.

También quiero dedicar dos palabras de agradecimiento al cuerpo docente de la UPV, por formarme y, después de 4 años de dedicación hacerme llegar hasta aquí.

He querido dejar en último lugar, para hacer una mención especial a mi tutor de TFG, Alberto José Pérez. Muchísimas gracias por haber sido tan buen profesor, profesional y humanamente, por prestarme el material necesario para replicar el proyecto, estar desde el primer día hasta el último siempre disponible pese a la distancia, contestar rápidamente a todas mis dudas y animarme en cada correo.

## BIBLIOGRAFÍA

---

[1] Visión por computador: qué es y cuáles son sus usos más comunes [en línea] [fecha de consulta: 22 marzo 2020]. Disponible en: <https://blog.infaimon.com/vision-computador-soluciones-permite/>

[2] Visión por computador [en línea] [fecha de consulta: 22 marzo 2020]. Disponible en: <https://www.hisour.com/es/computer-vision-42799/>

[3] Carlos, R.V. Sistemas robotizados. Grado Ingeniería Informática de la Universitat Politècnica de València. [PDF]. [fecha de consulta: 24 marzo 2020].

[4] Cámaras industriales: tipos y características. [en línea]. [fecha de consulta: 22 marzo 2020]. Disponible en: <https://blog.infaimon.com/camaras-industriales/>

[5] Cámaras inteligentes. [en línea]. [fecha de consulta: 24 marzo 2020]. Disponible en: <https://www.infaimon.com/categoria-producto/camaras/camaras-inteligentes/>

[6] Ángulo de visión. [en línea]. [fecha de consulta: 24 marzo 2020]. Disponible en: [https://es.wikipedia.org/wiki/%C3%81ngulo\\_de\\_visi%C3%B3n](https://es.wikipedia.org/wiki/%C3%81ngulo_de_visi%C3%B3n)

[7] La distancia focal explicada con ejemplos. [en línea]. [fecha de consulta: 27 marzo 2020]. Disponible en: <https://www.dzoom.org.es/la-distancia-focal-todo-lo-que-necesitas-saber-explicado-con-ejemplos/>

[8] Raspberry Pi. [en línea]. [fecha de consulta: 27 marzo 2020]. Disponible en: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)

[9] Raspberry Pi. [en línea]. [fecha de consulta: 30 marzo 2020]. Disponible en: <https://www.raspberrypi.org/>

[10] ¿Qué es Raspberry Pi? [en línea]. [fecha de consulta: 2 abril 2020]. Disponible en: <https://www.luisllamas.es/que-es-raspberry-pi/>

[11] Raspberry Pi 4 review: finally ready to replace your desktop PC [en línea]. [fecha de consulta: 5 abril 2020]. Disponible en: <https://www.wired.co.uk/article/raspberry-pi-4-review-price-release>

[12] ¿Qué es Arduino? [en línea]. [fecha de consulta: 9 abril 2020]. Disponible en: <https://arduino.cl/que-es-arduino/>

[13] Arduino Micro. [en línea]. [fecha de consulta: 9 abril 2020]. Disponible en: <https://store.arduino.cc/arduino-micro>

[14] Qué es Arduino, cómo funciona y qué puedes hacer con uno. [en línea]. [fecha de consulta: 12 abril 2020]. Disponible en: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

[15] Camera Module V2 . [en línea]. [fecha de consulta: 15 abril 2020]. Disponible en: <https://www.raspberrypi.org/products/camera-module-v2/>

[16] Raspberry Pi Touch Display. [en línea]. [fecha de consulta: 16 abril 2020]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-touch-display/>



[17] Pantalla oficial 7" Raspberry Pi multitáctil capacitiva. [en línea]. [fecha de consulta: 16 abril 2020]. Disponible en: <https://www.tiendatec.es/raspberry-pi/pantallas/31-pantalla-oficial-7-raspberry-pi-multitactil-capacitiva-5056070923850.html?src=raspberrypi>

[18] Processing. [en línea]. [fecha de consulta: 24 abril 2020]. Disponible en: <https://es.wikipedia.org/wiki/Processing>

[19] Processing. [en línea]. [fecha de consulta: 24 abril 2020]. Disponible en: <https://processing.org/>

[20] Arduino software. [en línea]. [fecha de consulta: 27 abril 2020]. Disponible en: <https://www.arduino.cc/en/Main/Software>

[21] Arduino IDE. [en línea]. [fecha de consulta: 27 abril 2020]. Disponible en: [https://es.wikipedia.org/wiki/Arduino\\_IDE](https://es.wikipedia.org/wiki/Arduino_IDE)

[22] OpenCV. [en línea]. [fecha de consulta: 3 mayo 2020]. Disponible en: <https://opencv.org/>

[23] OpenCV. [en línea]. [fecha de consulta: 3 mayo 2020]. Disponible en: <https://es.wikipedia.org/wiki/OpenCV>

[24] OpenGL. [en línea]. [fecha de consulta: 7 mayo 2020]. Disponible en: <https://es.wikipedia.org/wiki/OpenGL>

[25] OpenGL Overview. [en línea]. [fecha de consulta: 7 mayo 2020]. Disponible en: <https://www.opengl.org/about/>

[26] ControlP5. [en línea]. [fecha de consulta: 12 mayo 2020]. Disponible en: <http://www.sojamo.de/libraries/controlP5/>

[27] OpenCV for Processing. A creative coding computer vision library based on the official OpenCV Java API. [en línea]. [fecha de consulta: 13 mayo 2020]. Disponible en: <https://github.com/atduskgreg/opencv-processing>

[28] Resistencia pull up y pull down. [en línea]. [fecha de consulta: 15 mayo 2020]. Disponible en: <https://programarfacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/>

[29] Conceptos del color. [en línea]. [fecha de consulta: 18 mayo 2020]. Disponible en: <https://helpx.adobe.com/es/photoshop-elements/using/color.html>

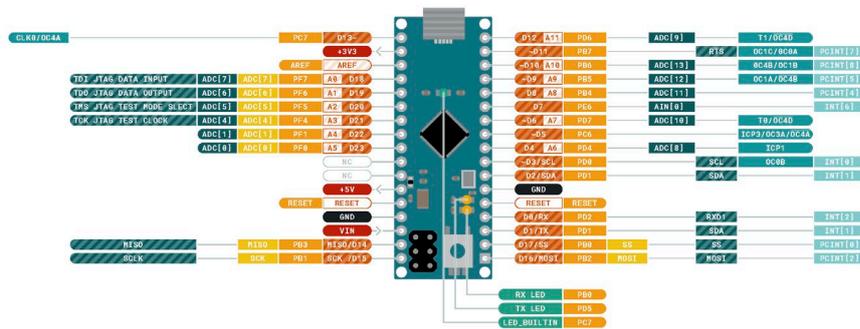


# ANEXOS

## DIAGRAMA GPIO ARDUINO MICRO



ARDUINO  
MICRO  
10141.ARDUINO.CO/MICRO



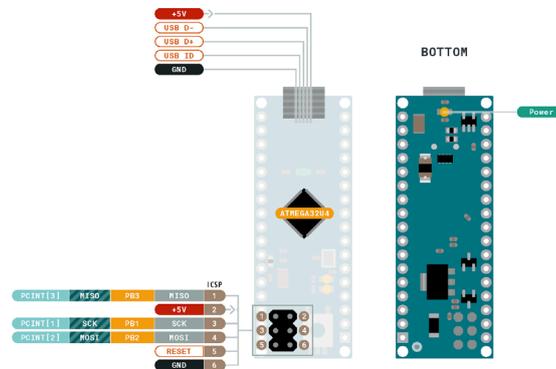
Ground	Digital Pin	Analog	<b>MAXIMUM</b> current per I/O pin is 40mA	<b>VIN</b> 6-20 V input to the board.
Power	Analog Pin	Communication	<b>MAXIMUM</b> current per +3.3V pin is 50mA	
LED	Other Pin	Timer		
Internal Pin	Microcontroller's Port	Interrupt		
SWD Pin	Default	Sercom		

ARDUINO . CC  
Last update: 23/03/2020

The work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94040, USA.

Figura anexos-1 Diagrama GPIO Arduino Micro, parte nº1. Imagen extraída de [13]





Ground	Digital Pin	Analog	<b>▲</b> MAXIMUM current per I/O pin is 40mA <b>▲</b> MAXIMUM current per +3.3V pin is 50mA	<b>VIN</b> 6-20 V input to the board.
Power	Analog Pin	Communication		
LED	Other Pin	Timer		
Internal Pin	Microcontroller's Port	Interrupt		
SWD Pin	Default	Sercom		

ARDUINO . CC  
 Last update: 23/03/2025

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94042, USA.

Figura anexos-2. Diagrama GPIO Arduino Micro, parte n°2. Imagen extraída de [13].

## CONFIGURACIÓN Y PRUEBA MODULO CÁMARA

Se recomienda tener precaución, ya que generalmente los módulos de las cámaras son sensibles a la electricidad estática, por lo tanto, antes de manejarlos se aconseja tocar una toma de tierra. Una vez tomadas todas las precauciones, conectar el módulo de la cámara al puerto CSI del ordenador Raspberry a través del cable plano.

Después de haber realizado la conexión, ejecutar las siguientes instrucciones en el terminal del ordenador para descargar e instalar la última versión disponible del kernel, del firmware GPU, y de las aplicaciones instaladas.

```
sudo apt update
sudo apt full-upgrade
```

El siguiente paso consiste en habilitar la cámara de nuestro ordenador, para ello se utiliza el programa de configuración preinstalado, digitando en el terminal el siguiente comando:

```
sudo raspi-config
```

Se muestra al usuario una ventana modal y utilizando las flechas direccionales del teclado se elige, en el listado que nos aparece, la voz “opciones de interfaz”. A continuación, se selecciona la voz “cámara” y se habilita siguiendo los pasos que nos muestra la ventana.

Una vez guardados los cambios realizados, el programa nos preguntará si se desea reiniciar el ordenador para que éstos se apliquen internamente. Confirmamos y esperamos que se vuelva a inicializar el sistema operativo.

La cámara ha sido habilitada, ahora sólo queda verificar que funciona correctamente. Esta comprobación se puede realizar de diferentes formas, la más rápida es utilizando una aplicación integrada en el sistema operativo de Raspberry Pi. Para ello, abrimos el terminal y digitamos la instrucción:

```
raspistill -v -o prueba.jpg
```

Este comando tiene el siguiente significado:

- Raspistill: Nombre del programa usado para probar la cámara.
- -v: Parámetro que especifica al programa qué debe mostrar durante la ejecución, mensajes de información sobre las operaciones que se realizan.
- -o: Parámetro que indica al programa en qué fichero se debe guardar la imagen obtenida. A continuación, se debe indicar el nombre de dicho fichero.
- prueba.jpg: Nombre del fichero donde se guarda la imagen capturada por la cámara.

Después de algunos segundos, si no aparece ningún aviso de error, el programa finaliza su ejecución y, en nuestro caso, si abrimos el directorio donde se ha ejecutado el orden, se encontrará el fichero correspondiente a la fotografía.

Si la cámara no se ha configurado correctamente y, por lo tanto, no está preparada para su uso, es probable que aparezcan durante la ejecución del programa los siguientes mensajes de error. A continuación, se describe brevemente su significado y la solución para resolver cada uno de ellos.



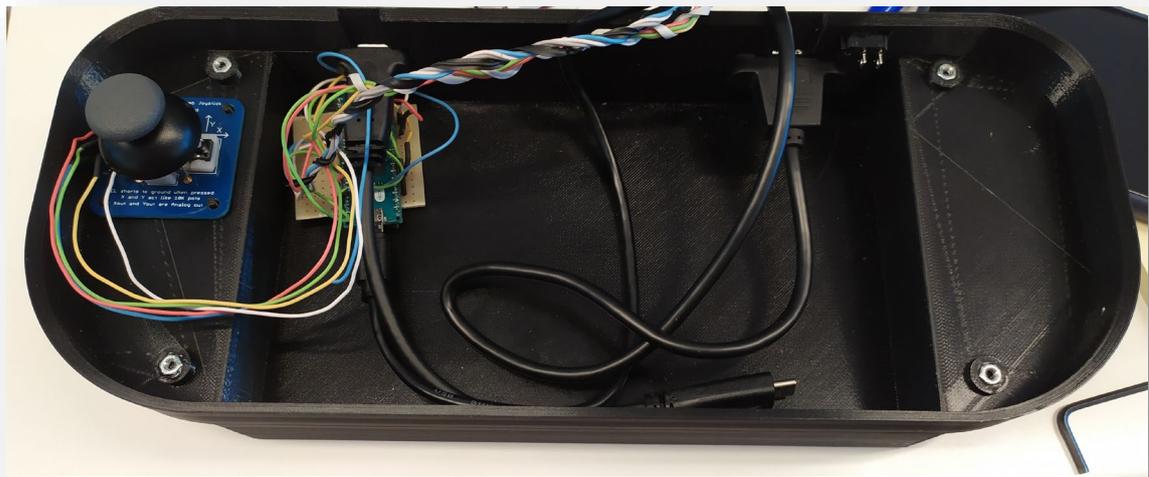
Mensaje de error	Significado	Solución
<b>raspistill/raspivid command not found</b>	Raspistill no se encuentra instalado. Significa que las operaciones de actualización y descarga, no se han ejecutado correctamente.	Realizar las operaciones de actualización una segunda vez.
<b>ENOEM</b>	La cámara no se está inicializando, probablemente hay alguna conexión defectuosa.	Se aconseja comprobar la conexión del cable plano.
<b>ENOSPC</b>	La cámara esta funcionando, pero la memoria de la GPU puesta a su disposición es inferior a la que necesita.	El parámetro gpu_mem del fichero config.txt en /boot/folder debe de ser por lo menos 128. Si no es así utilizar la opción Memoy Split de raspi-config para modificarlo.

Tabla 10. Errores comunes configuración cámara y sus respectivas soluciones.

## IMÁGENES MONTAJE Y FUNCIONAMIENTO DEL MANDO DE CONTROL



Figura anexos-3. Montaje parte superior del mando de control.



*Figura anexos-4. Montaje parte inferior mando de control.*



*Figura anexos-5. Mando de control operativo.*