

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Optimización de la herramienta de procesamiento de imágenes  
para el sistema Brainlab de HUMANA - Fase IV**

Protocolo de trabajo de graduación presentado por Sergio Alejandro  
Boch Ixén, estudiante de Ingeniería Mecatrónica

Guatemala,

2024

## Resumen

Este proyecto se enfoca en la optimización de la herramienta de procesamiento de imágenes y reconocimiento de caracteres utilizada en el sistema Brainlab de HUMANA, y su integración con un sistema robótico físico. Los objetivos específicos incluyen la revisión y mejora de algoritmos de procesamiento de imágenes, la validación con un mayor número de imágenes, la creación de un protocolo de comunicación entre el sistema de procesamiento de imágenes y el robot, y la mejora de la interfaz gráfica de usuario (GUI).

La metodología empleada abarca varias etapas, desde la validación de pruebas con imágenes adicionales, la optimización de algoritmos, el desarrollo de protocolo de comunicación, hasta la integración y optimización de la GUI. El proyecto culmina con la validación final de todo el sistema integrado, asegurando su precisión, fiabilidad y eficiencia.

Los resultados esperados incluyen una mejora significativa en la precisión del reconocimiento de caracteres, una interfaz de usuario más amigable y eficiente, y una integración exitosa con un sistema robótico, lo que permitirá aplicaciones prácticas en condiciones del mundo real.

## Antecedentes

El procesamiento digital de imágenes es un campo de continua evolución. El objetivo del procesamiento de imágenes puede tener a su vez distintos motivos. Algunos de estos podrían ser: Mejorar la calidad de imagen, conseguir una representación más eficiente que permita almacenarla en menor espacio sin una pérdida apreciable de calidad, extraer información relevante de cara a interpretar su contenido y tomar decisiones al respecto, etc.

## Investigaciones externas a la Universidad del Valle de Guatemala

En el trabajo de graduación de Alejandro Aranguren y Tiffany Vela [1] se presenta el diseño y desarrollo de un sistema de seguimiento de objetos mediante el procesamiento digital de imágenes aplicado al control de robots autónomos. El software permite al usuario seleccionar los colores de la marca que desea monitorear, además le permite cambiar los niveles de luminosidad con los que se presenta la imagen en pantalla. El sistema esta conformado por un robot autónomo capaz de realizar movimientos rotacionales y traslacionales de manera de evitar que la marca se pierda en algún momento. El computador posee un software desarrollado en el entorno MATLAB potenciado con tecnología de procesamiento paralelo a través de una tarjeta de video con tecnología CUDA para optimizar el procesamiento de imágenes que permite realizar la identificación del objeto de interés. Se logró controlar el movimiento del robot de tal manera que se tenga un movimiento suave y agradable a la vista. Es importante destacar que la utilización de una tarjeta de video GPU, acelera el procesamiento de imágenes, pero siempre y cuando este sea utilizado de la manera correcta, de otro modo el procesamiento puede resultar incluso más lento que antes. El diseño de un robot sencillo de dos grados de libertad, de poco peso y por ende intermedia simplicia el control del sistema y permite mayor precisión.

## HUMANA

El centro de Epilepsia y Neurocirugía Funcional HUMANA [2], es una organización formada por profesionales en Neurociencias que trabajan en beneficio de los pacientes que padecen problemas Neurológicos de difícil control, Epilepsia, Parkinson, Tumores Cerebrales, Columna Vertebral, Movimientos Anormales entre otros.

HUMANA está basada en un modelo de atención de categoría internacional con experiencia médica y uso de tecnología de vanguardia. En HUMANA se cuenta con unidades equipadas con cámara robótica y equipo de Neuroestimulación visual. Cuentan con un Quirófano diseñado y equipado para la cirugía cerebral y de columna, donde se cuenta con un neuronavegador para realizar procedimientos quirúrgicos con imágenes guiadas por computadora, por lo que la visión por computadora, procesamiento de imágenes y *machine learning* son conceptos que a HUMANA les genera interés y les son de utilidad.

## Brainlab

Brainlab [3], es una empresa pionera de la tecnología médica digital, explora e identifica nuevas posibilidades para desarrollar los mejores tratamientos posibles para los pacientes. En general, su tecnología puede posibilitar que las intervenciones quirúrgicas sean más eficientes, personalizadas y seguras. La tecnología utilizada los impulsa a combinar imágenes médicas y tecnologías innovadoras para garantizar que se aprovechan al máximo en beneficio de los pacientes.

Brainlab desarrolla un software donde correlaciona imágenes conformando un modelo digital del paciente que se optimiza y depura constantemente con ayuda de inteligencia artificial y sofisticados algoritmos. Mediante la combinación del mundo físico y digital es posible procesar y proporcionar información crítica de individualizada del paciente cuando se necesite.

Para lograr que el máximo número de pacientes se beneficien, Brainlab pone a disposición sus datos y tecnología básica a otras empresas, de este modo se logran conocimientos de distintas áreas clínicas para propiciar sinergias. En HUMANA utilizan el software de Brainlab para mostrar a los doctores encargados de operaciones del cerebro la posición y el ángulo en que tienen que insertar ciertas agujas. Dicho posicionamiento se realiza de forma manual, con un brazo mecánico. El proceso es lento y es lo que se busca automatizar con el software.

## Investigaciones en Universidad del Valle de Guatemala

En la Universidad del Valle de Guatemala se han realizado diversos estudios y proyectos donde involucran visión por computadora, procesamiento de imágenes y *machine learning*. Con el pasar de los años se ha buscado automatizar los procesos para hacerlos más eficientes y sencillos de utilizar para usuarios potenciales. Un ejemplo de este estudio es el trabajo realizado por E. Portales [4] donde se enfoca en la optimización de la herramienta de procesamiento de imágenes para el sistema Brainlab [3] en HUMANA [2].

La Universidad del Valle de Guatemala ha mantenido una colaboración de investigación con el Centro de Epilepsia y Neurocirugía Funcional HUMANA [2]. Durante este tiempo, se han llevado a cabo varios proyectos que buscan mejorar los procesos dentro del centro, abarcando desde la mejora en el control de los instrumentos hasta el procesamiento de los resultados de los diversos estudios realizados en ese lugar.

En la tesis de Josué Cifuentes [5] se enfoca en el diseño e implementación de un mando de control para una versión robotizada del sistema Varioguide de Brainlab. El proyecto se realizó en colaboración con el centro Neurológico HUMANA y la Universidad del Valle de Guatemala para mejorar la eficiencia en cirugías. Se dividió en tres partes. Primero el análisis de servomotores Dynamixel AX-12A y MX106T. El diseño estructural donde se creó el primer prototipo de la estructura mecánica y donde se descubrieron limitaciones computacionales y físicas. El diseño de mando de control donde se aplicó todo para controlar el brazo mecánico robotizado. El sistema Varioguide se ajusta por etapas, incluyendo ajustes tridimensionales, angulares y escalares. Los comandos de movimiento fueron satisfactorios, pero el problema que se presentó fue la estabilidad del sistema.

En la tesis de Santiago Galicia [6] se creó un algoritmo de *machine learning* para identificar dígitos de calibración en la pantalla del laboratorio de HUMANA. Se utilizó un modelo de red neuronal convolucional que logró una precisión del 98.6 % en el modelo y del 91.6 % en las pruebas físicas realizadas. Además, se diseñó un mecanismo de anclaje para un sensor óptico en la pantalla del laboratorio, lo que mejoró la efectividad y redujo el tiempo del sistema de calibración VarioGuide. Se implementó un sistema de procesamiento de imágenes para reconocer variables en la pantalla de la computadora, junto con un algoritmo para calificar cámaras y determinar la más adecuada para el reconocimiento de características necesarias. Además, se diseñó un formato TCP para enviar los datos de calibración leídos del sistema de Varioguide a un microcontrolador ESP32, facilitando la transferencia de información.

En la tesis de Estephan Portales [4] se trabajó en la optimización de algoritmos para el reconocimiento óptico de caracteres, centrándose en los ángulos proporcionados por el sistema de Brainlab utilizado por HUMANA. Se desarrolló una interfaz para facilitar el procesamiento e identificación de estos ángulos, junto con un sistema de servidor-cliente que conecta la computadora principal con un sistema embebido para recortar y transmitir imágenes.

El proyecto empleó el motor Tesseract como herramienta principal y Asprise OCR como herramienta secundaria para el reconocimiento de caracteres. Se realizaron pruebas preliminares con diferentes tipos de textos para verificar su funcionamiento. Además, se evaluó una tercera herramienta llamada OCR Space, aunque sus resultados no fueron satisfactorios.

Se diseñaron cuatro interfaces: dos para ejecutar todo el procesamiento en la misma interfaz en la computadora principal, utilizando Tesseract y Asprise OCR respectivamente, y otras dos para trabajar con el servicio de servidor-cliente implementado en el sistema embebido, cada una utilizando un motor diferente. Los resultados en este caso fueron satisfactorios realizando pruebas con un Arduino, sin embargo, no se comprobó con un robot real.

## Justificación

En el trabajo de graduación para el grado de licenciatura de Estephan Portales [4] y Santiago Galicia [6] se logró un avance significativo en la interfaz para el procesamiento de imágenes y en la herramienta para procesamiento óptico de caracteres con Tesseract. Sin embargo, los resultados obtenidos no fueron totalmente concluyentes, se tuvieron algunas limitaciones en cuanto a la calidad de imágenes por cámara web y en cuanto a la validación con el entorno físico, ya no se realizaron pruebas con un brazo robótico.

Existen métodos de optimización que fueron implementados de manera limitada o simplemente no fueron implementados en el sistema de procesamiento e interfaz de Estephan Portales [4], que pudieron haber hecho la interfaz más amigable y eficiente. Por lo anterior mencionado, en este trabajo se planteó la necesidad de desarrollar un sistema de captura de pantalla más eficiente que pueda cumplir con los requisitos de exactitud y confiabilidad validado con un brazo robótico. Así mismo, se busca la simplificación de la interfaz para que sea aun más amigable y eficiente para el público objetivo que estará utilizándola.

Validar el procesamiento de imágenes para ángulos con un sistema robótico físico es crucial para asegurar la precisión y exactitud del sistema, así como su integración adecuada. Estas pruebas permitirán verificar la funcionalidad, medir el rendimiento, detectar y corregir errores, y optimizar tanto los algoritmos de procesamiento como los controles del sistema robótico. Se busca garantizar que el sistema puede operar en condiciones del mundo real de manera confiable, confirmando su preparación para aplicaciones prácticas.

## Objetivos

### Objetivo General

Optimizar la herramienta de procesamiento de imágenes y reconocimiento de caracteres para el sistema Brainlab de HUMANA, y utilizar los resultados del reconocimiento de caracteres para el posicionamiento de un sistema robótico físico.

### Objetivos Específicos

- Revisar los algoritmos de procesamiento de imágenes desarrollados en fases anteriores con el fin de optimizar el reconocimiento de caracteres y mejorar su precisión.
- Validar la herramienta de procesamiento de imágenes y reconocimiento de caracteres desarrollada en la fase anterior con más imágenes, obtenidas con capturas de pantalla, del sistema Brainlab de HUMANA.
- Validar el protocolo de comunicación y envío de comandos propuesto en la fase anterior con un sistema robótico físico.
- Crear una interfaz de usuario fácil de usar para el manejo y configuración de la herramienta desarrollada.

## Marco teórico

### Imagen Digital

Una imagen se define como una función de dos dimensiones  $f(x,y)$  donde  $x$  e  $y$  son las coordenadas de un plano que contiene todos los puntos de la misma, y  $f(x,y)$  es la amplitud en el punto  $(x,y)$  a la cual se le llama intensidad o nivel de gris de la imagen en ese punto. En el caso de que tanto las coordenadas  $x$  e  $y$  como los valores de intensidad de la función  $f$  sean discretos y finitos, se habla de una imagen digital.

Una imagen digital está compuesta de un número finito de elementos y cada uno tiene una localidad y un valor particular. A estos elementos se les llama puntos elementales de la imagen o píxeles, siendo este último el término comúnmente utilizado para denotar la unidad mínima de medida de una imagen digital. [7]

### Análisis y procesamiento digital de imágenes

El Análisis Digital de Imágenes es el área de la ingeniería que se encarga de la extracción de mediciones, datos o información contenida en una imagen. Un sistema de análisis de imágenes se distingue debido a que tiene como parámetro de entrada una imagen, y cuyo resultado es comúnmente una salida numérica, en lugar de otra imagen. Esta salida es la información referente al contenido de la imagen de entrada.

Para llegar desde la imagen original al conjunto de parámetros e información extraída de la misma, es necesario pasar por distintas etapas de procesamiento y filtrado donde se analiza la imagen y se adecua para cierta aplicación específica.

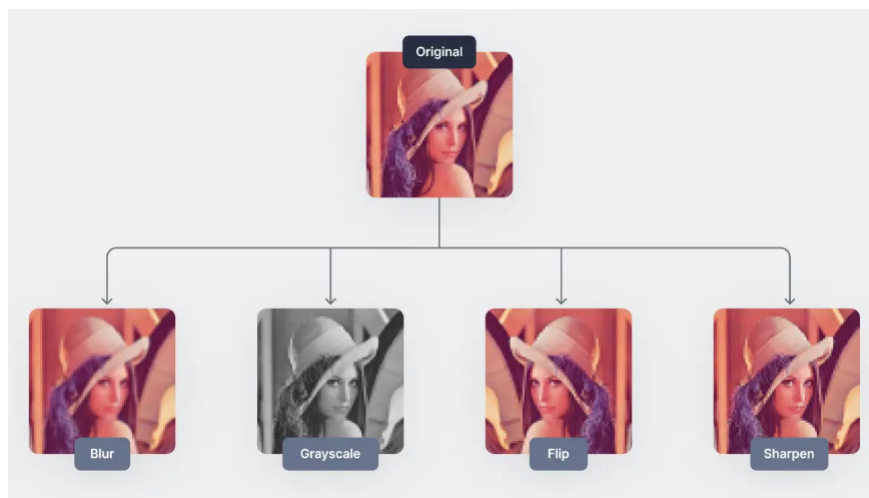


Figura 1: Ejemplos de procesamientos de imágenes típicos. [8]

Estas herramientas se organizan según el nivel de procesamiento que se desea realizar para analizar la información contenida en una imagen digital.

- **Pre-procesamiento.** Operaciones para adaptar la información de una imagen y tener mejor análisis en pasos posteriores. Ejemplos de procesamiento son las operaciones de brillo y contraste.
- **Segmentación.** Operaciones para hacer una partición de la imagen en varias regiones que representen la información necesaria para el problema a resolver.
- **Detección de objetos y clasificación.** Determinación y clasificación de objetos contenidos en la imagen.
- **Análisis de imagen.** Obtener información de alto nivel acerca de lo que la imagen muestra.

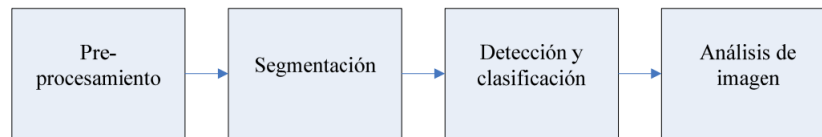


Figura 2: Niveles de procesamiento. [9]

El conjunto de métodos de procesamiento de imágenes esta dividido en tres grandes grupos:

- **Algoritmos en el dominio espacial.** Se refiere a métodos que procesan una imagen píxel por píxel, o también tomando en cuenta un conjunto de píxeles vecinos.
- **Algoritmos en el dominio de la frecuencia.** Frecuentemente, estos métodos son aplicados sobre los coeficientes resultantes de la Transformada de Fourier de una imagen.
- **Algoritmos de extracción de características.** A diferencia de los dos grupos anteriores, los algoritmos de extracción de características están enfocados al análisis de imágenes para la extracción de atributos y regiones de interés, separación de objetos del fondo, detección de bordes o formas, entre otros.

## Visión por computadora

Visión es la ventana al mundo de muchos organismos. Su función principal es reconocer y localizar objetos en el ambiente mediante el procesamiento de imágenes. La visión por computadora es el estudio de estos procesos, para entenderlos y construir máquinas con capacidades similares. [10]

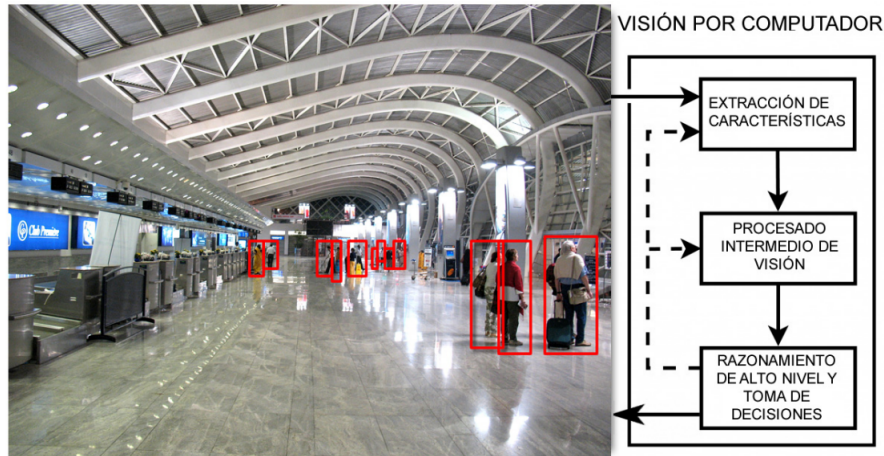


Figura 3: Visión por computadora y cadena de procesamiento típica. [7]

El objetivo de la visión por computadora es extraer características de una imagen para su descripción e interpretación por la computadora. Por ejemplo:

- Determinar la localización y tipo de objetos en la imagen.
- Construir una representación tridimensional de un objeto.
- Analizar un objeto para determinar su calidad.
- Descomponer una imagen u objeto en diferentes partes.

En visión se busca obtener descripciones útiles para cada tarea a realizar. La tarea demandará modificar ciertos atributos.

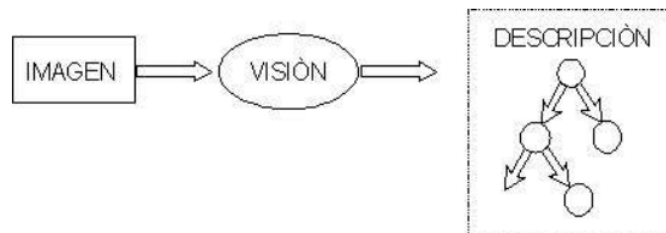


Figura 4: Esquema general de visión por computadora. [10]

Notesé que en la Figura 4 la imagen de entrada es procesada para extraer los atributos, obteniendo como salida una descripción de la imagen analizada.

## Python

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. Es ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el *machine learning*.



En cuanto a la realización de tareas y *machine learning* en Python, consiste en extraer conocimientos valiosos a partir de los datos, mientras que el enseña a las computadoras a aprender automáticamente de los datos y a efectuar predicciones precisas. [11]

## Open CV

Open CV es una librería de computación visual para el procesamiento de imágenes en Python. Esta biblioteca proporciona herramientas para realizar operaciones de procesamiento de imágenes, como el filtrado, la detección de bordes, el reconocimiento de características, el seguimiento de objetos, etc. Estas herramientas permiten desarrollar aplicaciones de visión artificial, como el reconocimiento facial, el seguimiento de objetos, etc.

## Reconocimiento óptico de caracteres (OCR)

La tecnología de reconocimiento de caracteres, OCR (Optical Character Recognition) engloba a un conjunto de técnicas basadas en estadísticas, en las formas de los caracteres, transformadas y en comparaciones, que complementándose entre sí, se emplean para distinguir de forma automática entre los diferentes caracteres alfanuméricos existentes.

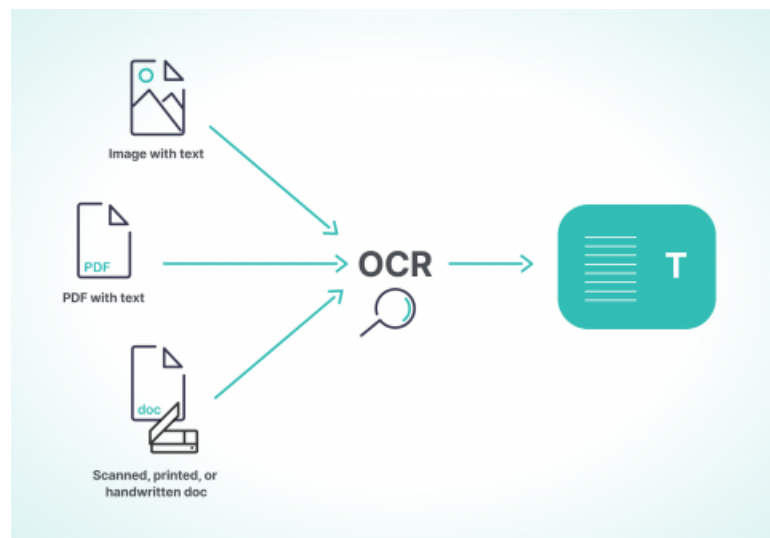


Figura 5: Diagrama de reconocimiento óptico de caracteres. [12]

En todo sistema de reconocimiento óptico de caracteres (OCR) se distinguen al menos estas 4 etapas:

- Adecuación de la imagen (preproceso).
- Selección de la zona de interés (segmentación).
- Representación digital de la imagen (extracción de características).

- Distinción del carácter contenido en la imagen (reconocimiento).

Para cada una de las cuatro etapas es posible aplicar multitud de técnicas ya existentes o desarrollar alguna específica en función de las condiciones en las que se representan los datos de entrada, que en el caso de OCR se puede traducir por las imágenes de entrada. [13]

## Tesseract

Tesseract es un motor OCR de código abierto que extrae texto impreso o escrito de las imágenes. Fue desarrollado originalmente por Hewlett-Packard, y su desarrollo fue adquirido después por Google. Con Tesseract OCR, es posible extraer texto de las imágenes con un reconocimiento eficaz de patrones y caracteres y en línea del motor de OCR.

Tesseract ya es capaz de reconocer más de 100 idiomas. Tesseract cuenta con una integración de IA a través de la Red Neuronal LSTM para detectar y reconocer mejor el contenido del texto de diferentes tamaños.

Uno de los aspectos principales de Tesseract es que es compatible con muchos lenguajes de programación y estructuras que utilizan wrappers como Pytesseract, también conocido como Python-Tesseract. [14]

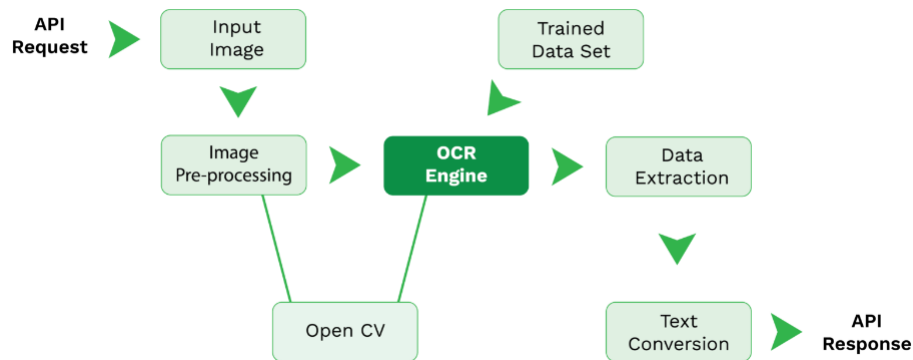


Figura 6: Fases del proceso Tesseract OCR. [14]

## Universal Asynchronous Receiver-Transmitter (UART)

UART significa receptor/transmisor asíncrono universal y definir un protocolo, o conjunto de reglas para intercambiar datos en serie entre dos dispositivos. El UART solo utiliza dos cables entre el transmisor y receptor para transmitir y recibir en ambas direcciones. La comunicación en UART puede ser simplex (los datos se envían en una sola dirección), semi-dúplex (cada lado transmite, pero solo uno a la vez), o dúplex completo (ambos lados pueden transmitir en simultáneo). Los datos en el UART se transmiten en la forma de tramas. [15]

El protocolo UART se utiliza ampliamente en una variedad de aplicaciones, desde la comunicación serie básica entre dispositivos embebidos hasta la interfaz con periféricos en

sistemas informáticos y de comunicaciones. Su simplicidad, flexibilidad y fiabilidad lo convierten en una opción popular para la interconexión de dispositivos electrónicos en numerosos contextos industriales y de consumo.

## Sistemas embebidos

Un sistema embebido posee hardware de computador junto con software embebido como uno de sus componentes más importantes. Es un sistema computacional dedicado para aplicaciones o productos. Puede ser un sistema independiente o parte de un sistema mayor, y dado que usualmente su software está embebido en ROM no necesita memoria secundaria como un computador. Un sistema embebido tiene tres componentes principales:

- Hardware
- Un software primario o aplicación principal. Este software o aplicación lleva a cabo una tarea en particular, o en algunas ocasiones una serie de tareas.
- Un software operativo que permite supervisar las aplicaciones, además de proveer los mecanismos para la ejecución de procesos. En muchos sistemas embebidos es requerido que el sistema operativo posea características de tiempo real.

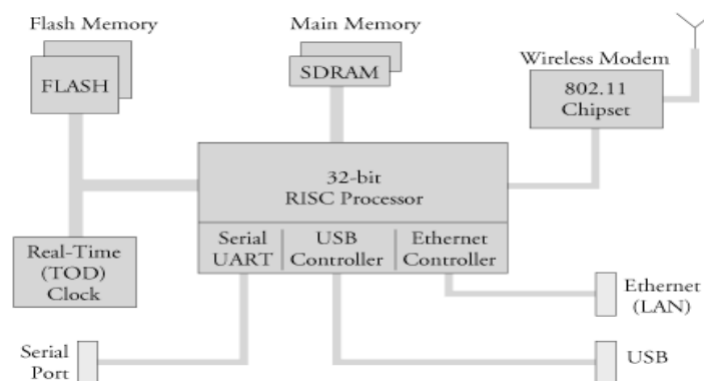


Figura 7: Ejemplo de un sistema embebido. [16]

## Sistemas robóticos

La robótica es una de las expresiones de la tecnología cuya aplicación se ha extendido a diversos contextos de la vida del hombre. La robótica, como tecnología que es, constituye el saber y hacer sobre los robots, esto implica el uso del conocimiento de diversas áreas para el diseño, construcción, ensamble y puesta en funcionamiento de un robot con un fin específico.

## Robótica en medicina

En medicina, algunas de las aplicaciones se encuentran en robots que son teleoperados por médicos especializados, ubicados en cualquier parte del mundo y quienes realizan la intervención quirúrgica con precisión microscópica. Entre los beneficios obtenidos cabe mencionar: disminución de costos por desplazamiento médico y la precisión en las acciones al eliminar ruidos como el temblor de las manos; en los desarrollos macrobióticos se resalta la construcción de prótesis que reemplazan eficientemente partes del cuerpo humano. [17]

### myCobot

myCobot es el robot colaborativo de seis ejes más pequeño y ligero del mundo. Se utiliza para aumentar la productividad y para el desarrollo secundario según las demandas de los usuarios.

myCobot es compacto pero potente; puede combinarse con una variedad de efectores finales para adaptarse a diferentes tipos de escenarios de aplicación. Además, admite el desarrollo secundario de software multiplataforma. Basado en diferentes tipos de aplicaciones, myCobot ofrece interfaces de código abierto que permiten el reconocimiento de objetos, reconocimiento facial, reconocimiento de imágenes, entre otros.

myCobot puede programarse en varios lenguajes de programación, como Python, C++, Arduino, C##, JS, etc. También ofrece diferentes métodos de conexión, como USB, Wi-Fi y Bluetooth. [18]



Figura 8: Robot myCobot. [18]

### OWI-007

El OWI-007, también conocido como *Robotic Arm Edge*, es un kit de construcción de brazo robótico diseñado para propósitos educativos y de entretenimiento. El brazo robótico puede ser controlado manualmente a través de un panel de control incluido, lo que permite a los usuarios experimentar con sus capacidades y aprender sobre principios de mecánica, electrónica y programación de robots.

El OWI-007 es utilizado en entornos educativos como una herramienta para enseñar conceptos de robótica y tecnología a estudiantes de todas las edades. Además, también se emplea en contextos de entretenimiento, donde los aficionados a la robótica pueden disfrutar construyendo y experimentando con este dispositivo. [19]



Figura 9: Robot OWI-007. [19]

## Metodología

### Validación de pruebas con más imágenes

- **Aumentar la robustez del sistema:** Ampliar el conjunto de datos de imágenes para asegurar que la herramienta de procesamiento de imágenes puede manejar una variedad de condiciones y casos atípicos, incrementando su robustez.
- **Mejorar la generalización:** Validar que la herramienta puede generalizar bien a nuevas imágenes no vistas anteriormente, asegurando que el sistema es aplicable en diversos contextos y situaciones.
- **Detectar y corregir errores:** Identificar posibles errores en el reconocimiento de caracteres al exponer el sistema a más datos, y desarrollar estrategias para corregir estos errores, mejorando así la precisión y confiabilidad del sistema.
- Recolección de capturas de pantalla adicionales para el sistema Brainlab de HUMANA.
- Implementación de pruebas, evaluando la capacidad del sistema para manejar diversas condiciones y variaciones en las imágenes, para validar la precisión y robustez del reconocimiento de caracteres en las nuevas imágenes.
- Análisis de los resultados para identificar posibles errores y áreas de mejora. Clasificar los errores en distintas categorías, como errores de reconocimiento (caracteres incorrectamente identificados), omisiones (caracteres no reconocidos) y falsos positivos (elementos identificados como caracteres que no lo son).

## Optimización de algoritmos de procesamiento

- **Mejorar la precisión:** Aumentar la precisión del reconocimiento de caracteres mediante la optimización de los algoritmos de procesamiento de imágenes, lo cual es crucial para aplicaciones críticas en el ámbito médico.
- **Reducir el tiempo de procesamiento:** Minimizar el tiempo necesario para procesar cada imagen, haciendo que el sistema sea más eficiente y adecuado para su uso en tiempo real.
- **Incorporar nuevas técnicas:** Implementar técnicas avanzadas de procesamiento de imágenes y aprendizaje automático para mejorar el rendimiento y adaptabilidad del sistema.
- Revisión de los algoritmos actuales desarrollados en fases anteriores.
- Implementación de optimizaciones utilizando técnicas avanzadas de procesamiento de imágenes y *machine learning*. Así mismo repetir las pruebas para validar que los cambios han mejorado el rendimiento del sistema.
- Comparar los nuevos resultados con los resultados originales para cuantificar la mejora en términos de las métricas definidas como precisión, sensibilidad, tiempo de respuesta, etc.

## Desarrollo del protocolo de comunicación con el sistema robótico

- **Establecer comunicación confiable:** Crear un protocolo de comunicación fiable entre la herramienta de procesamiento de imágenes y el sistema robótico para asegurar que los comandos y datos se transmiten de manera precisa y sin errores.
- **Facilitar la integración y expansión:** Diseñar un protocolo flexible y escalable que permita futuras expansiones del sistema y la integración con otros dispositivos o sistemas.
- **Optimizar la velocidad de comunicación:** Asegurar que la comunicación entre el procesamiento de imágenes y el sistema robótico es rápida y eficiente, permitiendo respuestas en tiempo real.
- Diseño de protocolo de comunicación utilizando estándares de transmisión de datos. Considerar diferentes estándares de transmisión como UART, I2C, SPI, TCP/IP, etc. Evaluar cada opción en términos de velocidad, fiabilidad, simplicidad, y compatibilidad con los dispositivos involucrados.
- Implementar el protocolo de comunicación en el entorno de desarrollo, asegurando que el sistema de procesamiento de imágenes y el sistema robótico puedan comunicarse eficazmente.
- Validar que el sistema de comunicación implementado funciona correctamente, permitiendo el envío y recepción de comandos entre el sistema de procesamiento de imágenes y el brazo robótico sin errores.

## Integración de la comunicación al robot a la GUI

- **Facilitar la interacción del usuario:** Crear una interfaz gráfica de usuario (GUI) intuitiva que permita a los usuarios interactuar fácilmente con el sistema, facilitando el manejo del robot y la herramienta de procesamiento de imágenes.
- **Asegurar la sincronización en tiempo real:** Garantizar que la GUI refleje en tiempo real los comandos y datos transmitidos entre el procesamiento de imágenes y sistema robótico.
- **Mejorar la experiencia del usuario:** Diseñar la GUI para ser lo más amigable posible, reduciendo la curva de aprendizaje y aumentando la eficiencia del usuario en la operación del sistema.
- Desarrollo de la GUI en Python utilizando bibliotecas como Tkinter o PyQt. Crear bocetos y prototipos de la GUI para planificar el diseño y la disposición de los elementos de la interfaz. Esto puede incluir el diseño de ventanas, menús y otros componentes interactivos, esto para encontrar la configuración más práctica de utilizar.
- Integración del protocolo de comunicación dentro de la GUI para permitir la interacción en tiempo real, esto implementando funciones que permitan al usuario enviar comandos desde la GUI al sistema robótico. Esto puede incluir la introducción de parámetros específicos y la ejecución de comandos al presionar botones.
- Validar que la GUI es intuitiva y funcional, proporcionando una experiencia de usuario positiva y facilitando la interacción con el sistema.

## Optimización de GUI con Python

- **Mejorar el uso:** Refinar la interfaz gráfica de usuario para que sea más intuitiva y fácil de usar, basada en el *feedback* de pruebas de uso con usuarios finales.
- **Incrementar la eficiencia del usuario:** Hacer que la GUI sea más eficiente en términos de tiempo y esfuerzo requerido para realizar tareas, mejorando así la productividad del usuario.
- **Asegurar la estabilidad y robustez:** Garantizar que la GUI es estable y robusta, minimizando errores y problemas durante su uso.
- Evaluar la interfaz gráfica de usuario para identificar problemas con el uso y áreas de mejora, asegurando una interacción eficiente y satisfactoria para los usuarios.
- Realizar mejoras en la GUI en función del *feedback* de los usuarios y de las pruebas de funcionalidad para optimizar la experiencia del usuario y la operatividad del sistema.
- Refinar continuamente la GUI a través de ciclos iterativos de evaluación y mejora, garantizando que la interfaz sea cada vez más eficiente y fácil de usar.

## Validación del sistema integrado

- **Asegurar la funcionalidad completa:** Validar que todo el sistema, tanto como el procesamiento de imágenes, reconocimiento de caracteres, protocolo de comunicación y sistema robótico, funcionen correctamente y de manera integrada en un entorno real.
- **Medir el rendimiento del sistema:** Evaluar el rendimiento del sistema en términos de tiempo de respuesta, precisión y fiabilidad para asegurarse de que cumple con los requisitos del proyecto.
- **Identificar y corregir fallos:** Detectar cualquier fallo o inconsistencia en el sistema integrado y realizar las correcciones necesarias para optimizar su funcionamiento.
- Crear casos de prueba que cubran todos los escenarios posibles, desde las condiciones normales hasta situaciones límite. Esto incluye la identificación y procesamiento de imágenes, evaluando la capacidad del sistema para identificar y analizar correctamente los caracteres en diferentes condiciones.
- Evaluar el rendimiento global el sistema mediante métricas clave como el tiempo de respuesta, la precisión y la fiabilidad, asegurando que el sistema cumple con los requisitos de desempeño esperados.
- Detectar y corregir errores en el sistema para optimizar su rendimiento, asegurando que el sistema funcione de manera eficiente y fiable en todas las condiciones de operación.



Cronograma de actividades

No.	Nombre de la tarea	Fecha de inicio	Fecha de finalización	Junio			Julio			Agosto			Septiembre						
				Semana 1: 2/6/2024 - 8/6/2024	Semana 2: 9/6/2024 - 15/6/2024	Semana 3: 16/6/2024 - 22/6/2024	Semana 4: 23/6/2024 - 29/6/2024	Semana 5: 30/6/2024 - 6/7/2024	Semana 6: 7/7/2024 - 13/7/2024	Semana 7: 14/7/2024 - 20/7/2024	Semana 8: 21/7/2024 - 27/7/2024	Semana 9: 28/7/2024 - 3/8/2024	Semana 10: 4/8/2024 - 10/8/2024	Semana 11: 11/8/2024 - 17/8/2024	Semana 12: 18/8/2024 - 24/8/2024	Semana 13: 25/8/2024 - 31/8/2024	Semana 14: 1/9/2024 - 7/9/2024	Semana 15: 8/9/2024 - 14/9/2024	Semana 16: 15/9/2024 - 21/9/2024
1	Funcionamiento de myCobot																		
1.1	Interacción y uso de myCobot	2/6/2024	3/9/2024																
1.2	Instalación de librerías para programar en Python	2/6/2024	2/6/2024																
1.3	Comunicación y envío de datos a myCobot	4/6/2024	11/6/2024																
1.4	Evaluar posibles posiciones de manera manual	3/6/2024	12/6/2024																
2	Recolección y validación de datos																		
2.1	Recolección de imágenes adicionales	2/6/2024	15/6/2024																
2.2	Validación de la herramienta	3/6/2024	18/6/2024																
2.3	Evaluación de posibles errores y áreas de mejora	19/6/2024	21/6/2024																
2.4	Desarrollo de estrategias para corregir errores vistos	19/6/2024	3/7/2024																
3	Optimización de algoritmos																		
3.1	Revisión de algoritmos actuales	3/6/2024	28/6/2024																
3.2	Mejorar precisión de algoritmos	29/6/2024	16/7/2024																
3.3	Reducir el tiempo de procesamiento	4/7/2024	9/7/2024																
3.4	Incorporación de nuevas técnicas	10/7/2024	25/7/2024																
3.5	Pruebas de los nuevos algoritmos y las mejoras implementadas	28/6/2024	25/7/2024																
3.6	Comparación de resultados optimizados con los de fase previa	2/7/2024	25/7/2024																
4	Desarrollo de protocolo de comunicación																		
4.1	Revisión y evaluación de comunicación de fase previa	2/6/2024	15/6/2024																
4.2	Diseño de protocolo de comunicación	17/7/2024	29/7/2024																
4.3	Implementación del protocolo en el entorno de desarrollo	17/7/2024	15/8/2024																
4.4	Pruebas de validación del envío y recepción de comandos	16/8/2024	20/8/2024																
5	Interfaz gráfica																		
5.1	Evaluación de puntos de mejora en la GUI de fase previa	3/6/2024	25/6/2024																
5.2	Integración del protocolo de comunicación a la GUI	28/6/2024	7/9/2024																
5.3	Desarrollo de la interfaz gráfica con mejoras	28/6/2024	14/9/2024																
5.4	Pruebas de la nueva interfaz	28/6/2024	14/9/2024																
6	Validación del sistema integrado																		
6.1	Validar que todo el sistema funcione correctamente	8/9/2024	21/9/2024																
6.2	Evaluar el rendimiento del sistema integrado	15/9/2024	21/9/2024																
6.3	Identificación de errores y áreas de mejora	1/9/2024	28/9/2024																
7	Redacción de documento de tesis	2/6/2024	28/9/2024																

Figura 10: Cronograma de actividades

# Índice preliminar

- Prefacio
- Lista de figuras
- Resumen
- Abstract

## 1. Introducción

## 2. Antecedentes

- a)* Investigaciones externas a la Universidad del Valle de Guatemala
- b)* HUMANA
- c)* Brainlab
- d)* Investigaciones en Universidad del Valle de Guatemala

## 3. Justificación

## 4. Objetivos

- a)* Objetivo General
- b)* Objetivos Específicos

## 5. Alcance

## 6. Marco Teórico

- a)* Imagen Digital
- b)* Análisis y procesamiento digital de imágenes
- c)* Visión por computadora
- d)* Python
- e)* Open CV
- f)* Reconocimiento óptico de caracteres (OCR)
- g)* Tesseract
- h)* Universal Asynchronous Receiver-Transmitter (UART)
- i)* Sistemas embebidos
- j)* Sistemas robóticos
- k)* Robótica en medicina
- l)* myCobot
- m)* OWI-007

## 7. Metodología

- a)* Validación de pruebas con más imágenes

- b) Optimización de algoritmos de procesamiento
  - c) Desarrollo del protocolo de comunicación con el sistema robótico
  - d) Integración de la comunicación al robot a la GUI
  - e) Optimización de GUI con Python
  - f) Validación del sistema integrado
- 8. Visión por computadora
  - a) Optimización de algoritmos
  - b) Mejora en reconocimiento de caracteres
- 9. Protocolo de comunicación
  - a) UART
  - b) TCP/IP
  - c) Pruebas preliminares con protocolo
- 10. Interfaz Gráfica
  - a) Herramientas implementadas en la interfaz
  - b) Comunicación con el sistema robótico
  - c) Optimización de GUI con todo el sistema integrado
  - d) Resultado de envío y recepción de datos
- 11. Comunicación con sistema robótico
  - a) Pruebas preliminares
  - b) Estructura de envío de datos hacia el sistema robótico
  - c) Envío y recepción de datos
- 12. Conclusiones
- 13. Recomendaciones
- 14. Bibliografía

## Referencias

- [1] A. Aranguren y T. Vela, “Sistema de seguimiento de objetos mediante procesamiento digital de imágenes aplicado al control de robots autónomos,” Tesis de licenciatura, Universidad Peruana de Ciencias Aplicadas, 2012.
- [2] *HUMANA GT*, <https://humanagt.org/>, Accedido el 15 de marzo de 2024.
- [3] *Brainlab*, <https://www.brainlab.com/es/acerca-de-brainlab/>, Accedido el 15 de marzo de 2024.
- [4] F. E. Portales, “Optimización de la herramienta de procesamiento de imágenes para el sistema Brainlab de HUMANA - Fase III,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2023.

- [5] J. J. Cifuentes, “Diseño e implementación del mando de control para brazo robótico asistencial en cirugía estereotáctica del cerebro,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2019.
- [6] S. S. Galicia, “Optimización de la herramienta de procesamiento de imágenes para el sistema Brainlab de HUMANA,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
- [7] N. Aguirre, “Procesamiento de imágenes,” Proyecto de fin de carrera, Universidad Universidad de Sevilla, 2012.
- [8] *Image Processing: Techniques, Types, Aplications*, 2023. dirección: <https://www.v7labs.com/blog/image-processing-guide>.
- [9] R. Q. Juan y C. M. Mario, “Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década,” *RIIE&C, Revista de Ingeniería Eléctrica, Electrónica y Computación*, vol. 9, n.º 1, págs. 7-16, 2011.
- [10] L. E. Sucar y G. Gómez, “Visión computacional,” *Instituto Nacional de Astrofísica, Óptica y Electrónica. México*, 2011.
- [11] *¿Qué es Python?* Extraído de Amazon Web Series del apartado de herramientas para desarrolladores, 2023. dirección: <https://aws.amazon.com/es/what-is/python/>.
- [12] *Optical Character Recognition (OCR)*, 2024. dirección: <https://www.egnyte.com/guides/governance/optical-character-recognition>.
- [13] S. Fernández, C. Javier y V. S. Consuegra, “Reconocimiento óptico de caracteres (ocr),” *Univ. Carlo*, vol. 3, n.º 7, pág. 2008, 2008.
- [14] *Tesseract OCR: ¿Qué es y por qué lo deberías elegir en el 2024?* 2024. dirección: <https://www.klippa.com/es/blog/informativo/que-es-tesseract-ocr/>.
- [15] *¿Qué es UART?* 2024. dirección: [https://www.rohde-schwarz.com/es/productos/test-y-medida/essentials-test-equipment/digital-oscilloscopes/que-es-uart\\_254524.html#:~:text=UART%20\(universal%20asynchronous%20receiver%20%2F%20transmitter,y%20recibir%20en%20ambas%20direcciones..](https://www.rohde-schwarz.com/es/productos/test-y-medida/essentials-test-equipment/digital-oscilloscopes/que-es-uart_254524.html#:~:text=UART%20(universal%20asynchronous%20receiver%20%2F%20transmitter,y%20recibir%20en%20ambas%20direcciones..)
- [16] D. Pérez, “Sistemas embebidos y sistemas operativos embebidos,” *Lecturas en ciencias de la computación. Universidad Central de Venezuela, Vols. % i de % 2ISSN*, págs. 1316-6239, 2009.
- [17] P. A. L. Ramírez y H. A. Sosa, “Aprendizaje de y con robótica, algunas experiencias,” *Revista Educación*, págs. 43-63, 2013.
- [18] *myCobot: 6-axis collaborative Robotic Arm*, 2024. dirección: <https://www.elephantrobotics.com/en/mycobot-en/>.
- [19] *Robotic Kits: Robotic Arm Edge*, 2024. dirección: <https://owirobot.com/>.