configuración del módulo de la cámara y capturar imágenes.

**Paso 1: Configuración del Hardware de la Cámara**

-Compra de la Cámara:

Elegir una cámara compatible con el sistema, como una Raspberry Pi Camera Module para un proyecto basado en Raspberry Pi.

-Conexión de la Cámara:

Conectar la cámara al puerto CSI de la Raspberry Pi.

**Paso 2: Instalación del Software Necesario**

-Actualización del Sistema (Raspberry Pi):

sudo apt-get update

sudo apt-get upgrade

Habilitación de la Cámara (Raspberry Pi):

Ejecuta sudo raspi-config

Seleccionar “Interface Options” y habilitar la cámara.

-Instalación de OpenCV:

pip install opencv-python

**Paso 3: Captura de Imágenes**

-Script para Capturar Imágenes:

import cv2

# Inicializa la cámara

cap = cv2.VideoCapture(0) # Usa 0 para la cámara por defecto

if not cap.isOpened():

print("No se puede abrir la cámara")

exit()

# Captura una sola imagen

ret, frame = cap.read()

if not ret:

print("No se puede recibir el frame (stream end?). Exiting ...")

exit()

# Muestra el frame capturado

cv2.imshow('frame', frame)

# Guarda la imagen capturada

cv2.imwrite('captura.jpg', frame)

# Espera a que el usuario presione una tecla

cv2.waitKey(0)

# Libera la cámara

cap.release()

cv2.destroyAllWindows()

**Paso 4: Preprocesamiento de Imágenes**

-Redimensionamiento y Normalización:

Redimensiona las imágenes a un tamaño uniforme (por ejemplo, 128x128 píxeles) y normaliza los valores de los píxeles.

# Leer la imagen

img = cv2.imread('captura.jpg')

# Redimensionar la imagen

img\_resized = cv2.resize(img, (128, 128))

# Normalizar la imagen

img\_normalized = img\_resized / 255.0

-Conversión a Tensor (si usas TensorFlow):

import tensorflow as tf

# Convertir a tensor

img\_tensor = tf.convert\_to\_tensor(img\_normalized, dtype=tf.float32)

# Añadir una dimensión para el lote

img\_tensor = tf.expand\_dims(img\_tensor, axis=0)

**Paso 5: Integración con el Sistema**

-Automatización de la Captura y Preprocesamiento:

import time

while True:

# Captura y preprocesa la imagen

ret, frame = cap.read()

img\_resized = cv2.resize(frame, (128, 128))

img\_normalized = img\_resized / 255.0

img\_tensor = tf.convert\_to\_tensor(img\_normalized, dtype=tf.float32)

img\_tensor = tf.expand\_dims(img\_tensor, axis=0)

# Aquí podrías enviar la imagen procesada a tu modelo de IA

# modelo.predict(img\_tensor)

time.sleep(60) # Espera 60 segundos antes de la siguiente captura