



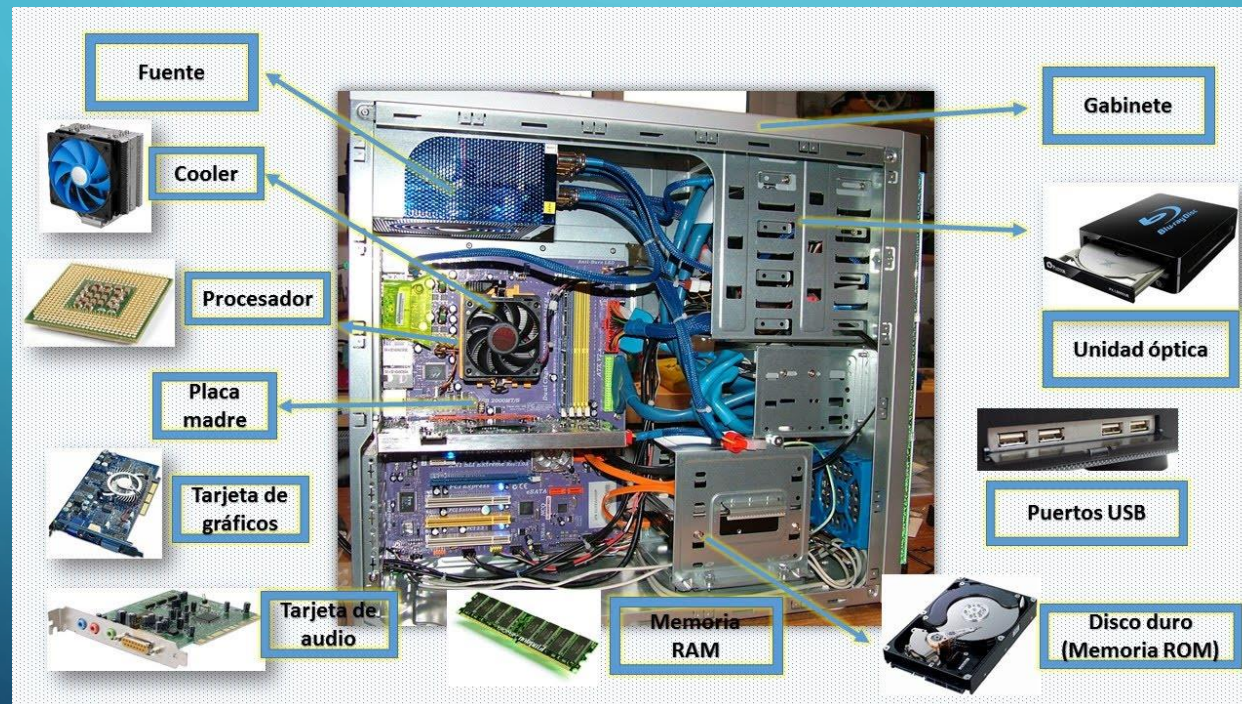
PRESENTACIÓN

SONG LAGUNA ERIC ALEJANDRO

Asignatura	Taller de computo
Año o semestre en que imparte	1ero
Horas clase a la semana	4 horas
Unidad	Unidad 2: Hardware y software
Aprendizajes	Generaciones de computadoras: Explica la evolución del hardware y software de la computadora

LA PROBLEMÁTICA

- Se dará a conocer los diferentes componentes que tiene una computadora y como estos interactúan en su funcionamiento.



JUSTIFICACIÓN

El programa de Python es un lenguaje sencillo de aprendizaje el cual dará un primer acercamiento a lo que es la programación.



PRODUCTO ESPERADO

- Los alumnos y alumnas entregaran un mapa donde se vean los diferentes componentes de una PC y su respectivo funcionamiento.
- También los alumnos y las alumnas entregaran un pseudocódigo de un programa en Python sencillo que haga una operación matemática.
- Después interactuaran con un programa hecho en Python que identifica los componentes.
- Por ultimo los alumnos y las alumnas, crearan un pseudocódigo que haga uso de uno de los periféricos de la computadora (cámaras, puertos USB, teclado, etc..), esto para mostrar como pueden interactuar a través de los códigos hechos por uno mismo.

PRESENTACIÓN DEL PROBLEMA A RESOLVER

- El alumno observara la evolución de la computadora y como esta afecta su entorno, además de identificar como el lenguaje de programación en Python puede ser una herramienta con la cual puede crear diferentes programas con ejemplos y el apoyo del docente.



Inicio de la Sesión

1. El **docente junto con los alumnos** realizará un repaso de los conocimientos previos.
2. El **docente realizara una explicación** sobre como los lenguajes de programación son una parte esencial del nuestro día a día y como este se integra a diferentes dispositivos que usamos a diario.
3. **Los alumnos y alumnas de manera individual** describirán el hardware de acuerdo con su función y características.

Desarrollo de la sesión

4. **Los alumnos y alumnas en equipo** investigaran las características de los distintos sistemas operativos, así como algunos comandos y operaciones comunes.
5. **Los alumnos y alumnas en equipo** explicaran las características de propiedad, distribución y libertad de uso del software.
6. **Se preparará un debate sobre los siguientes temas:** riesgos en el uso de software y daños en la salud por el uso de hardware.
7. **Los alumnos y alumnas** realizaran un diagrama que identifique los riesgos del uso de software y explica los daños que con lleva el hardware.
8. **Los alumnos y las alumnas** interactuaran con un programa que identifique objetos hecho en Python.

Cierre de la sesión

9. **Los alumnos y alumnas en equipo** sacaran sus conclusiones sobre el cómo tener buenas prácticas tanto en el uso del software como el del hardware.

¿POR QUE SE REQUIERE QUE CONOZCAN?

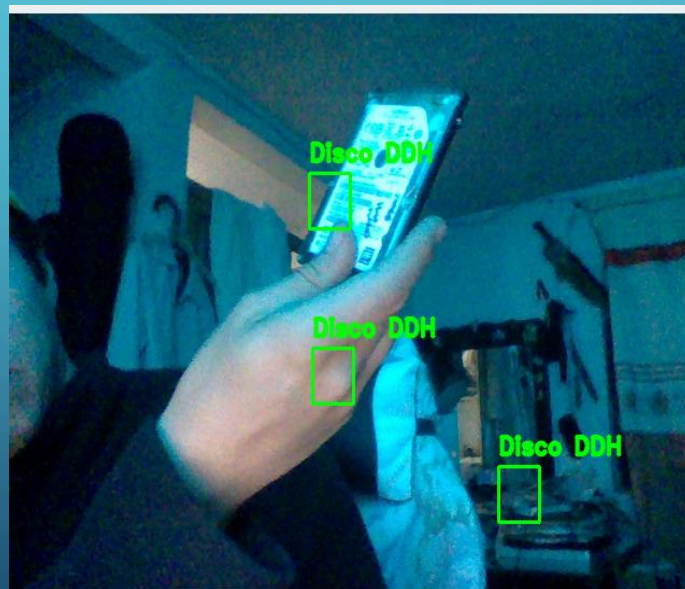


INCLUSIVE PUEDE SER UN INGRESO



PROGRAMA DE RECONOCIMIENTO

- *El proyecto de la detección de objetos por medio de la cámara web, se centra en que los alumnos pudiesen hacer uso de una herramienta interactiva la cual detecta a través del programa piezas de la computadora que les servirá de referencia para poder aprender y analizar su uso cotidiano*



- A continuación, se muestra algunas pantallas del programa

```
In [1]: ## Paquetes a instalar ##  
pip install os  
pip install cv2  
pip install imutils
```

```
In [ ]: ## Bibliotecas a usar para este programa ##  
  
import cv2  
import numpy as np  
import imutils  
import os  
  
## Esta parte del programa creara una carpeta donde guardaremos los resultados positivos  
  
Datos = 'p'  
if not os.path.exists(Datos):  
    print('Carpeta creada: ',Datos)  
    os.makedirs(Datos)          ## nos apoyamos de makedirs para crear  
  
Datos2 = 'n'  
if not os.path.exists(Datos2):  
    print('Carpeta creada: ',Datos2)  
    os.makedirs(Datos2)        ## nos apoyamos de makedirs para crear  
  
## A partir de aqui se iniciara el programa para la captura usando la biblioteca cv2, el count hara que  
## se genere un contador para las imagenes tomadas  
  
cap = cv2.VideoCapture(0)  
x1, y1 = 190, 80  
x2, y2 = 450, 398  
count = 0
```

PROGRAMA DE CAPTURA

```
## Aqui se creara un rectangulo que delimitara el area de nuestro modelo para obtener los resultados

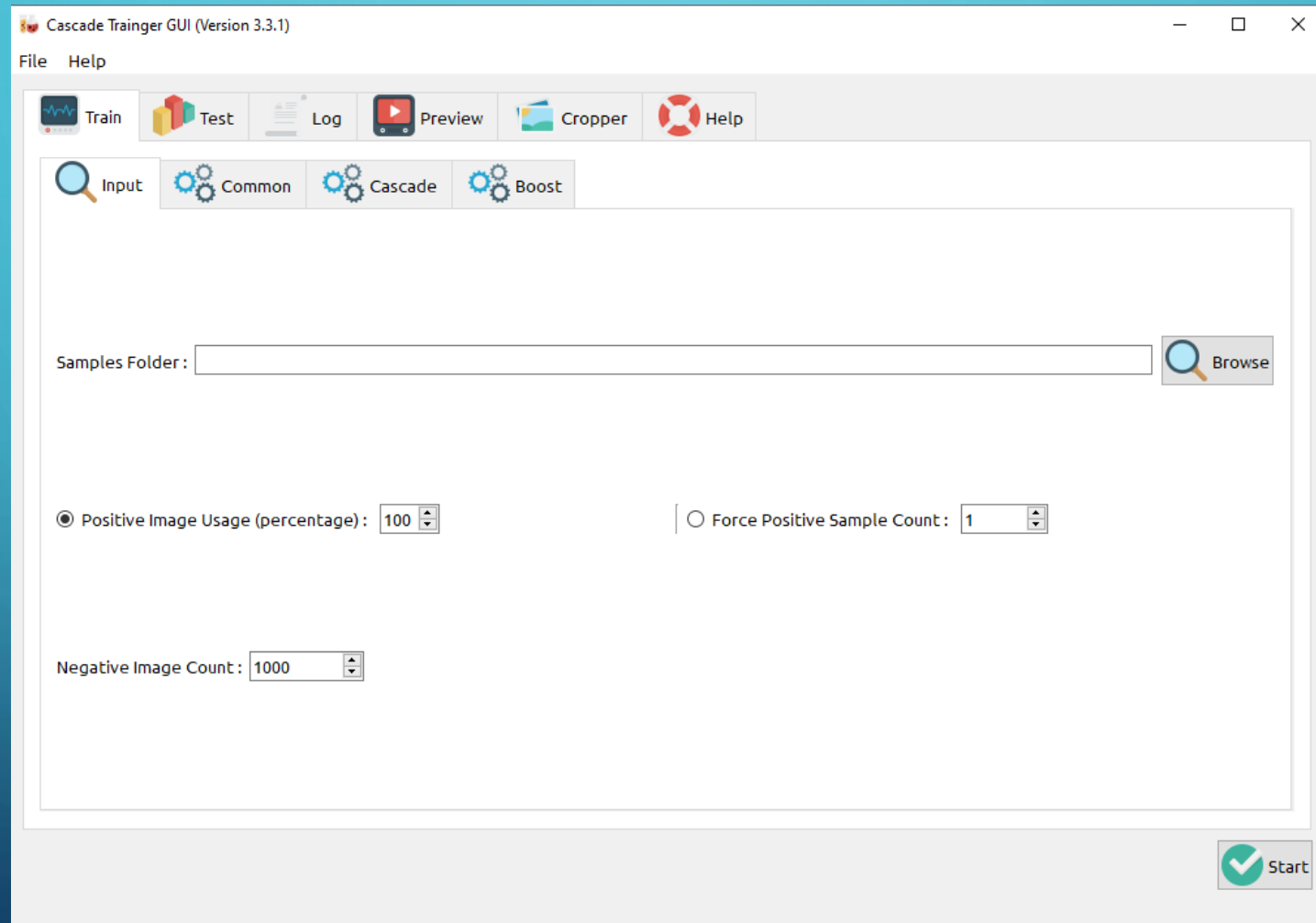
while True:
    ret, frame = cap.read()
    if ret == False: break
    imAux = frame.copy()
    cv2.rectangle(frame, (x1,y1), (x2,y2), (255,0,0), 2)
    objeto = imAux[y1:y2,x1:x2]
    objeto = imutils.resize(objeto, width=38)
    #print(objeto.shape)

## Esta parte guardara la imagen que tomemos una vez que la tengamos frente a la camara, se guardara con
## la tecla "s"

    k = cv2.waitKey(1)
    if k == ord('s'):
        cv2.imwrite(Datos+'/objeto_{}.jpg'.format(count), objeto)
        print('Imagen guardada: '+'/objeto_{}.jpg'.format(count))
        count = count + 1
    if k == 27:
        break

    cv2.imshow('frame', frame)
    cv2.imshow('objeto', objeto)
cap.release()
cv2.destroyAllWindows()
```


PROGRAMA DE ENTRENAMIENTO CASCADE



PROGRAMA DE RECONOCIMIENTO

```
In [1]: ## Paquete necesario ##
import cv2

## Aqui se usara la linea para el uso de la camara

cap = cv2.VideoCapture(0)

Clasificador = cv2.CascadeClassifier('cascade.xml') ## llamamos el archivo de entrenamiento
## El programa que se utilizo para entrenar el archivo cascade genero este archivo a partir de las fotos tomadas

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    toy = Clasificador.detectMultiScale(gray, scaleFactor = 10, minNeighbors = 350) ##Aqui se guardaran las detecciones
    ##scaleFactor y minNeighbors son los parametros de tolerancia

    for (x,y,w,h) in toy: ## Esto genera las coordenadas del rectangulo que va a detectar
        cv2.rectangle(frame, (x,y), (x + w, y + h), (0,255,0), 2)
        cv2.putText(frame, 'Disco DDH', (x,y - 10), 2, 0.7, (0, 255, 0), 2, cv2.LINE_AA) ## Genera el nombre del

    cv2.imshow('Captura', frame)

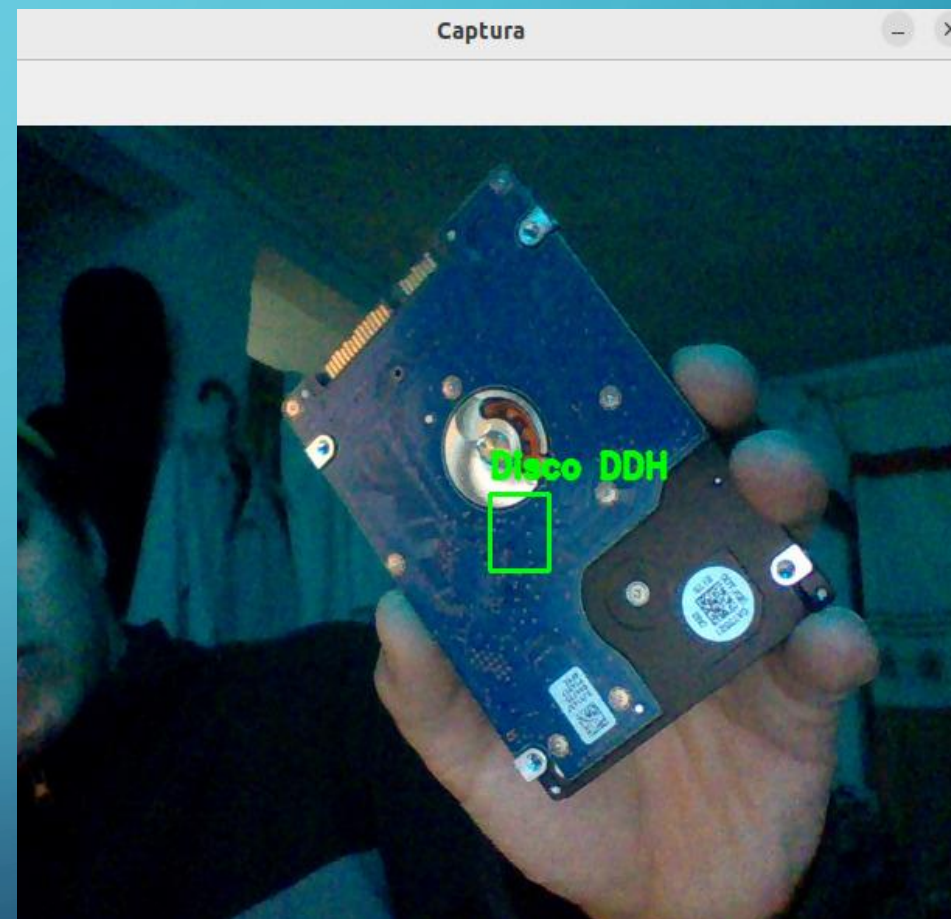
    if cv2.waitKey(1) & 0xFF == ord('q'): ## Para cerrar la pantalla con la letra 'q'
        break

cap.release()
cv2.destroyAllWindows()
```

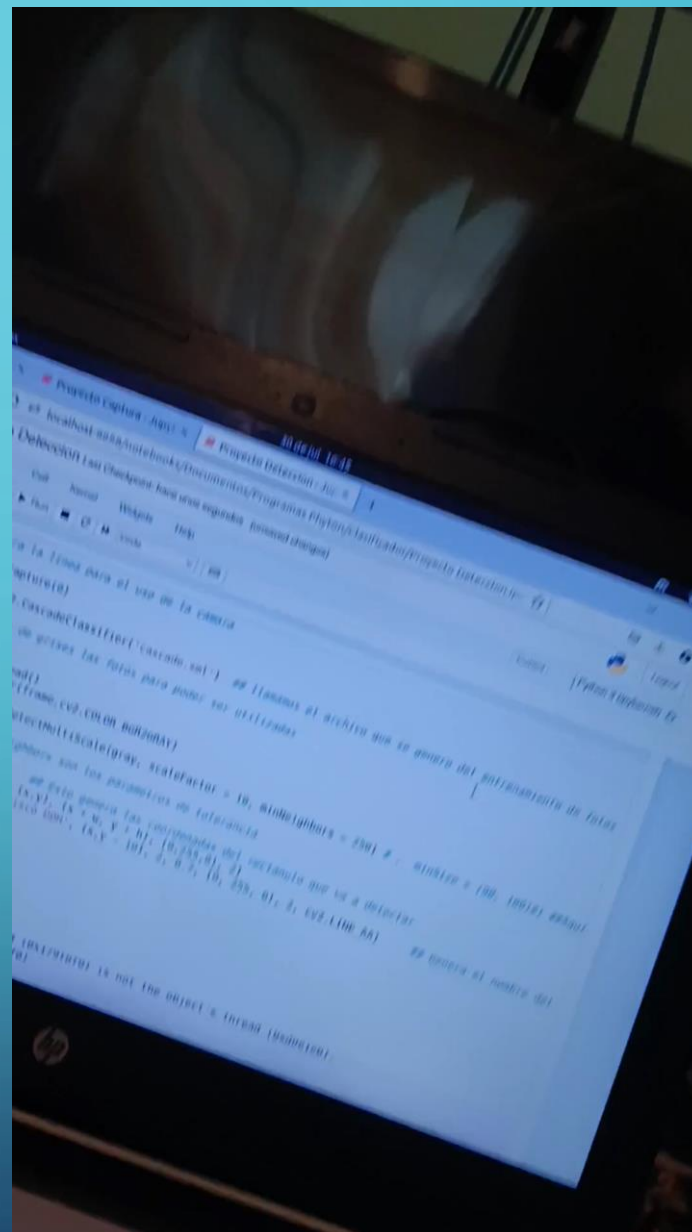
MUESTRAS VS RESULTADOS

Carpeta personal / Documentos / Programas Phyton / Clasificador / n

Nombre	Tamaño	Modificación
objeto_0.jpg	1.6 kB	15:08 ☆
objeto_1.jpg	1.6 kB	15:08 ☆
objeto_2.jpg	1.6 kB	15:08 ☆
objeto_3.jpg	1.5 kB	15:08 ☆
objeto_4.jpg	1.5 kB	15:08 ☆
objeto_5.jpg	1.6 kB	15:08 ☆
objeto_6.jpg	1.6 kB	15:08 ☆
objeto_7.jpg	1.5 kB	15:08 ☆
objeto_8.jpg	1.4 kB	15:08 ☆
objeto_9.jpg	1.4 kB	15:08 ☆
objeto_11.jpg	1.5 kB	15:08 ☆
objeto_12.jpg	1.6 kB	15:08 ☆
objeto_13.jpg	1.6 kB	15:08 ☆
objeto_14.jpg	1.5 kB	15:08 ☆



Este fue el primer video con una muestra de 350 imágenes de error contra 50 de acierto



Referencias

Tutorial de Python: <https://docs.python.org/es/3/tutorial/>
<https://j2logo.com/python/tutorial/>

Partes de una PC: <https://www.superprof.mx/blog/partes-pc/>

Series de partes: <https://www.adslzone.net/reportajes/tecnologia/detectar-componentes-hardware-pc/>

Comandos CMD: <https://www.xataka.com/basics/comandos-basicos-para-dar-tus-primeros-pasos-consola-windows-cmd>