

# **FORMACIÓN PROFESIONAL**

## **CURSO DE PRÁCTICA INTENSIVA**

### **CUADERNO DE INFORMES**



## INSTRUCCIONES PARA EL USO DEL CUADERNO DE INFORMES

### 1. PRESENTACIÓN.

El Cuaderno de Informes es un documento de auto control, en el cual el estudiante, registra diariamente, durante la semana, las tareas, operaciones que ejecuta en su aprendizaje, es un medio para desarrollar la Competencia de Redactar Informes.

### 2. INSTRUCCIONES PARA EL USO DEL CUADERNO DE INFORMES.

2.1 En la hoja de informe semanal, el estudiante registrará diariamente los trabajos que ejecuta, indicando el tiempo correspondiente. El día de asistencia registrará los contenidos que desarrolla. Al término de la semana totalizará las horas.

De las tareas ejecutadas durante la semana, el ESTUDIANTE seleccionará la tarea más significativa (1) y él hará una descripción del proceso de ejecución con esquemas, diagramas y dibujos correspondientes que aclaren dicho proceso.

2.2 Semanalmente, el Instructor revisará y calificará el Cuaderno de Informes haciendo las observaciones y recomendaciones que considere convenientes, en los aspectos relacionados a la elaboración de un Informe Técnico (letra normalizada, dibujo técnico, descripción de la tarea y su procedimiento, normas técnicas, seguridad, etc.

2.3 Escala de calificación vigesimal:

CUANTITATIVA	CUALITATIVA	CONDICIÓN
16,8 – 20,0	Excelente	Aprobado
13,7 – 16,7	Bueno	
10,5 – 13,6	Aceptable	
00 – 10,4	Deficiente	Desaprobado

## INFORME SEMANAL

.....V.....SEMESTRE    SEMANA N° 11

	<b>DIA</b>	<b>MES</b>	<b>AÑO</b>
<b>DEL</b>	<b>19</b>	<b>05</b>	<b>25</b>
<b>AL</b>	<b>23</b>	<b>05</b>	<b>25</b>

<b>DÍA</b>	<b>TAREAS EFECTUADAS</b>	<b>HORAS</b>
LUNES	Conocer el entorno de Google Colab.	6
MARTES		
MIÉRCOLES	Usar recursos y archivos de Google Colab.	6
JUEVES	Usar recursos y archivos de Google Colab.	6
VIERNES		
SÁBADO		
<b>TOTAL</b>		<b>18</b>

## INFORME DE TAREA MÁS SIGNIFICATIVA

### Tarea:

Realizar un sistema de detección de objetos en Python utilizando Google Colab.

### Descripción del proceso:

En este informe se detallará los pasos para la implementación de un sistema de detección de objetos personalizados utilizando el modelo YOLOv11 sobre la plataforma Google Colab. El sistema fue entrenado con un conjunto de datos personalizado y luego utilizado para detectar objetos en un archivo de video.

**Objetivo:** Implementar un sistema de detección de objetos personalizados.

#### ❖ Tecnologías utilizadas:

- a. **Google Colab.**
- b. **Python.**
- c. **Git/GitHub:** Control de versiones.

- ❖ **Carga del modelo base YOLOv11:** Se importa un modelo preentrenado YOLOv11, el cual ha sido entrenado previamente en grandes conjuntos de datos como COCO. Este modelo base contiene pesos y configuraciones que permiten detectar objetos genéricos, y sirve como punto de partida para ser ajustado (fine-tuned) con nuevos datos personalizados. Su uso permite acelerar el entrenamiento y mejorar la precisión cuando se desea detectar objetos específicos no incluidos en el entrenamiento original.
- ❖ **Entrenamiento del modelo personalizado:** Adaptar el modelo YOLOv11 a un conjunto de datos personalizado mediante un proceso de aprendizaje supervisado. Se proporcionan imágenes etiquetadas con las clases de objetos deseadas, y a través de múltiples iteraciones (épocas), el modelo ajusta sus pesos internos para reconocer patrones específicos de esos objetos. El objetivo de este entrenamiento es lograr que el modelo identifique correctamente los objetos personalizados con alta precisión y bajo error.
- ❖ **Hacer predicciones:** Aplicar el modelo entrenado sobre nuevos fotogramas para identificar y localizar los objetos aprendidos, devolviendo como salida las clases detectadas, sus coordenadas en la imagen y el nivel de confianza de cada detección.
- ❖ **Detectar objetos usando el modelo entrenado:** Aplicamos el modelo previamente entrenado sobre nuevos datos visuales (como imágenes o videos) con el objetivo de identificar los objetos personalizados aprendidos durante la fase de entrenamiento.

## HACER ESQUEMA, DIBUJO O DIAGRAMA

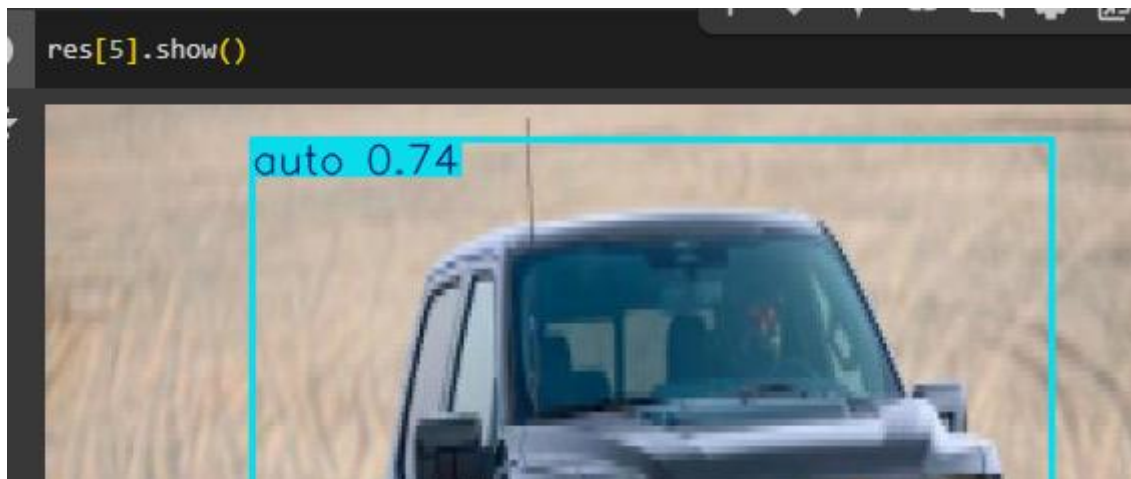
### Carga del modelo YOLOv11

```
model = YOLO('yolo11s.pt')
```

Downloading <https://github.com>

### Hacer predicciones

```
res = custom_model.predict(source='/content/valid')  
  
image 1/9 /content/deteccionObjetos-1/valid  
image 2/9 /content/deteccionObjetos-1/valid  
image 3/9 /content/deteccionObjetos-1/valid  
image 4/9 /content/deteccionObjetos-1/valid
```



### Detectar objetos usando el modelo entrenado

```
while cap.isOpened():  
    ret, frame = cap.read()  
    if not ret:  
        break  
  
    # Realizar inferencia  
    results = model(frame)[0]  
    annotated_frame = results.plot()  
  
    # Mostrar resultados  
    cv2.imshow("YOLO Inference", annotated_frame)  
  
    # Presiona ESC para salir  
    if cv2.waitKey(1) & 0xFF == 27:  
        break
```



Enlace de GitHub

<https://github.com/yusmer-edu/machineiA.git>

**EVALUACIÓN DEL INFORME DE TRABAJO SEMANAL**

NOTA

--

**OBSERVACIONES Y RECOMENDACIONES**

DEL INSTRUCTOR:


FIRMA DEL ESTUDIANTE:

FIRMA DEL INSTRUCTOR:

--	--



**PROPIEDAD INTELECTUAL DEL SENATI. PROHIBIDA SU  
REPRODUCCIÓN Y VENTA SIN LA AUTORIZACIÓN  
CORRESPONDIENTE**