



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE GUAYAQUIL**  
**CARRERA DE ELECTRÓNICA Y**  
**AUTOMATIZACIÓN**

**DESARROLLO E IMPLEMENTACIÓN DE UN MODELO DE MACHINE LEARNING  
DE GOOGLE PARA LA CLASIFICACIÓN DE OBJETOS POR COLOR**

Trabajo de titulación previo a la obtención del  
Título de Ingeniera en Electrónica

**AUTOR:** CRUZ SUÁREZ NOEMÍ AZUCENA

**TUTOR:** ING. LARCO TORRES VÍCTOR DAVID, MGTR

Guayaquil-Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE  
TITULACIÓN**

Yo, Noemi Azucena Cruz Suarez con documento de identificación N° 0919640615 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, febrero del año 2024

Atentamente,



---

Noemi Azucena Cruz Suarez

0919640615

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE  
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Cruz Suarez Noemi Azucena con documento de identificación No. 0919640615, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: "Desarrollo e implementación de un modelo de machine Learning de Google para la clasificación de objetos por color", el cual ha sido desarrollado para optar por el título de Ingeniero en Electrónico y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, febrero del año 2024

Atentamente,



---

Noemi Azucena Cruz Suarez

0919640615

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Larco Torres Víctor David con documento de identificación N° 0923270136, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO E IMPLEMENTACIÓN DE UN MODELO DE MACHINE LEARNING DE GOOGLE PARA LA CLASIFICACIÓN DE OBJETOS POR COLOR realizado por Noemi Azucena Cruz Suarez con documento de identificación N° 0919640615, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, febrero del año 2024

Atentamente,



---

Ing. Larco Torres Víctor David, Msc.

C.I. 0923270136

## **DEDICADO A**

*A Dios por permitirme vivir, por ser mi guía durante esta etapa de mi vida, por brindarme sabiduría, salud, paciencia y constante durante este proceso.*

*A mis padres, motor de mi vida y siempre inspiran a no rendirme y seguir luchando hasta alcanzar mis metas y mis sueños.*

*A mis hermanos, que me han animado, aconsejado y apoyado en todo sentido en cada etapa de mi carrera.*

*A mis sobrinos que me inspira a seguir adelante en verlos crecer para no bajar los brazos y continuar estudiando.*

*A mis docentes, que fueron un pilar por su apoyo constante en la carrera, me impulsaron a trabajar con mucho esfuerzo y perseverancia para culminar esta etapa universitaria.*

**Noemí Cruz S.**

## **AGRADECIMIENTO**

A Dios por bendecirme para alcanzar esta meta, a mis padres Alba Suárez y José Sanga, que ha sido la piedra fundamental de mis éxitos, pilares e inspiración de mi superación, por haberme forjado hacer mejor persona, una buena hija a través de sus consejos, enseñanzas, amor, su espera a mi regreso y por su bendición a diario después de cada salida.

A mi hermano Msc. Christian Sanga Suárez quién siempre me ha apoyado en todo este proceso de superación.

A mis amigos y compañeros que me han apoyado en mi Carrera.

A todo el personal administrativo, mantenimiento y docentes quienes me han dado su apoyo incondicional desde mis inicios y durante mi Carrera universitaria con el deseo de verme crecer como una gran profesional, y de manera especial a mi tutor de tesis Msc. Víctor Larco Torres y quién me animó a seguir esta Carrera universitaria desde que llegue a la Universidad y al Director de Carrera Msc. Orlando Barcia por su ayuda, sus cuidados, paciencia y sus conocimientos brindados.

Y al final a mi porque, aunque no fue fácil, pude lograrlo.

**Noemí Cruz S.**

## Resumen

Este escrito presenta el desarrollo y la implementación de un modelo de aprendizaje automático de Google para la clasificación de objetos por colores.

El paso inicial consistió en entrenar el modelo de cubo de color con la herramienta Teachable Machine de Google para el reconocimiento de colores. Una vez entrenado el cubo, se procedió al análisis en tiempo real y posteriormente a la clasificación.

En cuanto al sistema de recursos lógicos, se utilizó el editor P5j para alojar el modelo entrenado a partir de la aplicación Teachable Machine de Google después de realizar varias pruebas para obtener un modelo casi perfecto en función de las variables que se deben tener en cuenta a la hora de entrenar las muestras.

### **Palabras claves:**

Inteligencia Artificial, Machine Learning, Teachable machine, Visión Artificial, P5.js.

## Abstract

This document presents the development and implementation of a Google machine learning model for the classification of objects by color.

The initial step was to train the color cube model with Google's Teachable Machine tool for color recognition. Once the cube was trained, the analysis proceeded in real time and then the classification was performed.

As for the logical resources system, P5j editor was used to host the model trained from the Google Teachable Machine application after several tests to obtain an almost perfect model according to the variables that must be considered when training the samples.

### **Keywords:**

Artificial Intelligence, Machine Learning, Teachable Machine, Artificial Vision, P5.js.

## INDICE

<b>1.</b>	<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>2.</b>	<b>PROBLEMA.....</b>	<b>1</b>
<b>3.</b>	<b>OBJETIVOS .....</b>	<b>2</b>
<b>3.1.</b>	<b>OBJETIVO GENERAL.....</b>	<b>2</b>
<b>3.2.</b>	<b>OBJETIVO ESPECÍFICO.....</b>	<b>2</b>
<b>4.</b>	<b>FUNDAMENTOS TEÓRICOS.....</b>	<b>3</b>
<b>4.1.</b>	<b>INTELIGENCIA ARTIFICIAL.....</b>	<b>3</b>
<b>4.2.</b>	<b>VISIÓN ARTIFICIAL.....</b>	<b>3</b>
<b>4.3.</b>	<b>MACHINE LEARNING.....</b>	<b>4</b>
<b>4.4.</b>	<b>MACHINE LEARNING E INDUSTRIA 4.0.....</b>	<b>4</b>
<b>4.5.</b>	<b>TEACHABLE MACHINE.....</b>	<b>4</b>
<b>4.5.1.</b>	<b>VALORES PREDETERMINADOS PARA LA PREPARACIÓN DEL MODELO EN LA HERRAMIENTA DE TEACHABLE MACHINE.....</b>	<b>5</b>
<b>4.5.1.1.</b>	<b>ÉPOCAS.....</b>	<b>5</b>
<b>4.5.1.2.</b>	<b>MUESTRAS DE PREPARACIÓN.....</b>	<b>6</b>
<b>4.5.1.3.</b>	<b>TAMAÑO DE LOTE .....</b>	<b>6</b>
<b>4.5.1.4.</b>	<b>TAZA DE APRENDIZAJE.....</b>	<b>6</b>
<b>4.5.2.</b>	<b>GRÁFICA DE NIVEL DE RENDIMIENTO DE MODELO EN LA HERRAMIENTA DE TEACHABLE MACHINE.....</b>	<b>7</b>
<b>4.5.2.1.</b>	<b>PRECISIÓN POR CLASE .....</b>	<b>7</b>



4.5.2.2.	MUESTRAS DE PRUEBA.....	8
4.5.2.3.	MATRIZ DE CONFUSIÓN .....	8
4.5.2.4.	PRECISIÓN DE ÉPOCA .....	9
4.5.2.5.	PÉRDIDA DE ÉPOCA .....	9
4.5.3.	EXPORTAR MODELO .....	10
4.6.	P5.JS WEB EDITOR .....	10
5.	MARCO METODOLÓGICO .....	11
5.1.	PASOS PARA HACER UN PROYECTO DE IMAGEN EN LA HERRAMIENTA TEACHINE MACHINE EN GOOGLE.....	11
5.1.1.	INICIAR SESIÓN EN TEACHABLE MACHINE.....	12
5.1.2.	OBTENCIÓN DE LOS FOTOGRAMAS O IMÁGENES PARA LAS CLASES.....	13
5.1.3.	GUARDAR IMÁGENES EN EL PROYECTO .....	14
5.1.4.	PASOS PARA OBTENER EL MODELO .....	14
5.1.4.1.	PASOS PARA EXPORTAR MODELOS .....	20
5.1.5.	CLASIFICACIÓN DE OBJETOS POR COLOR POR MEDIO DE LA HERAMIENTA P5.js WEB EDITOR.....	22
6.	RESULTADOS.....	27
7.	CRONOGRAMA.....	31
8.	PRESUPUESTO.....	32
9.	CONCLUSIONES.....	32
10.	RECOMENDACIONES.....	32
11.	REFERENCIA BRIBLIOGRÀFICA.....	33

<b>12. ANEXOS .....</b>	<b>34</b>
<b>12.1. PROGRAMACIÓN.....</b>	<b>34</b>
<b>12.1.1. CÓDIGO DEL INDEX EN P5.js .....</b>	<b>34</b>
<b>12.1.2. CÓDIGO DEL SKTCH EN P5.js .....</b>	<b>35</b>

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Estructura para la clasificación de objetos por color.....	2
Ilustración 2 Brazo robótico seleccionando objetos por color (Cueto, 2022) .....	3
Ilustración 3 Proceso de creación del modelo con la herramienta .....	5
Ilustración 4 Valor de época.....	5
Ilustración 5 Tamaño de lote .....	6
Ilustración 6 Tasa de aprendizaje en la preparación del modelo.....	7
Ilustración 7 Precisión por clase.....	7
Ilustración 8 Matriz de confusión.....	8
Ilustración 9 Precisión de época.....	9
Ilustración 10 Precisión por clase y Pérdida por clase .....	10
Ilustración 11 Exportar modelo desde Teachable Machine.....	10
Ilustración 12 Herramienta P5.js .....	11
Ilustración 13 Ingreso a la página de Teachable Machine.....	12
Ilustración 14 Clases de muestras .....	12
Ilustración 15 Obtención de imágenes por medio de la cámara del computador .....	13
Ilustración 16 Visualización para obtener imágenes por medio de archivos .....	13
Ilustración 17 Guardar imagen o proyecto .....	14
Ilustración 18 Proyectos en Google Drive.....	14

Ilustración 19 Precisión por clases .....	15
Ilustración 20 Matriz de confusión.....	16
Ilustración 21 Precisión por época .....	17
Ilustración 22 Pérdida por época.....	17
Ilustración 23 Reconocimiento de la clase Otro en Teachable Machine por cámara del ordenador .....	18
Ilustración 24 Reconocimiento de la clase Azul en Teachable Machine por cámara del ordenador .....	19
Ilustración 25 Reconocimiento de la clase Rojo en Teachable Machine por cámara del ordenador ....	19
Ilustración 26 Reconocimiento de la clase Verde en Teachable Machine por cámara del ordenador ...	20
Ilustración 27 Botón de exportar modelo .....	20
Ilustración 28 Subir modelo .....	21
Ilustración 29 Modelo en nube y enlace para compartir .....	21
Ilustración 30 Copiado de URL de modelo.....	21
Ilustración 31 Fragmentos de código para usar modelo.....	22
Ilustración 32 Herramienta de web editable de P5.js .....	22
Ilustración 33 Autoguardar proyecto en P5.js.....	22
Ilustración 34 Índice de la herramienta de P5.js .....	23
Ilustración 35 Link del modelo en la herramienta de P5.js .....	23
Ilustración 36 Designa etiqueta de video y almacenar .....	24
Ilustración 37 Función de precarga .....	24
Ilustración 38 Inicio del código: Función setup .....	25
Ilustración 39 Función draw.....	25
Ilustración 40 Función de clasificación.....	26
Ilustración 41 Obtener resultado y error.....	26
Ilustración 42 Index de la herramienta de P5.js .....	27
Ilustración 43 Prototipo con Webcam apagada.....	27
Ilustración 44 URL en la herramienta P5.js .....	28
Ilustración 45 Reconocimiento de la clase Bandal en P5.js por Webcam.....	28

Ilustración 46 Reconocimiento de la clase Verde en P5.js por Webcam .....	29
Ilustración 47 Reconocimiento de la clase Azul en P5.js por Webcam .....	29
Ilustración 48 Reconocimiento de la clase Rojo en P5.js por Webcam .....	29
Ilustración 49 Reconocimiento de la clase Otro en P5.js por Webcam.....	30

## **ÍNDICE TABLAS**

Tabla 1 Cronograma de actividades .....	31
Tabla 2 Tabla de presupuesto.....	32

## **1. INTRODUCCIÓN**

En este contexto, el presente trabajo se enfoca en la implementación de un sistema de clasificación de imágenes por color utilizando técnicas de aprendizaje automático. El objetivo principal es desarrollar un modelo que pueda identificar y clasificar objetos según su color con un alto grado de precisión y robustez. Para ello, se empleará una combinación de herramientas de captura de imágenes, procesamiento de datos y algoritmos de aprendizaje automático.

El proceso de desarrollo incluirá la captura de fotogramas de colores de objetos utilizando una cámara conectada a un computador, el entrenamiento del modelo utilizando muestras de imágenes etiquetadas, y la evaluación del rendimiento del modelo mediante pruebas y validación cruzada. Se explorarán diferentes enfoques de preprocesamiento de datos, selección de características y ajuste de parámetros para optimizar el desempeño del sistema.

El resultado final será un sistema funcional capaz de reconocer y clasificar objetos por color en tiempo real, con aplicaciones potenciales en industrias como la manufactura, logística, seguridad y tecnología de consumo. Este trabajo contribuirá al avance en el campo de la visión por computadora y la inteligencia artificial aplicada, proporcionando una herramienta versátil y precisa para la identificación automatizada de objetos basada en el color.

## **2. PROBLEMA**

El desarrollo y la aplicación de sistemas de clasificación de imágenes por color han cobrado una importancia significativa en diversas áreas, desde la automatización industrial hasta la identificación de objetos en aplicaciones móviles.

Estos sistemas permiten identificar y categorizar objetos en función de sus características cromáticas, lo que facilita una amplia gama de aplicaciones prácticas. Sin embargo, el desafío radica en entrenar modelos de aprendizaje automático capaces de realizar esta tarea de manera eficiente y precisa.

Con este modelo se pretende mostrar cómo se lleva a cabo el proceso de reconocimiento y clasificación de cajas de 6cm x 5cm x 3cm por medio de una webcam como se muestra en la ilustración 1. Desde el punto de vista industrial, se pretende que este prototipo pueda observar el proceso por estudiantes y profesionales afines; ya que los módulos de aprendizaje se centran en el aprendizaje y la enseñanza, el proyecto es ejemplar y se enfoca en mejorar los procesos de las personas actualmente en las industrias.



*Ilustración 1 Estructura para la clasificación de objetos por color.*

### **3. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

- Clasificar objetos por color mediante un modelo de Machine Learning de Google.

#### **3.2. OBJETIVO ESPECÍFICO**

- Capturar los fotogramas de colores de las cajas y enviarlos a la herramienta Teachable Machine.
- Obtener el modelo de aprendizaje automático y cargar un computador.
- Clasificar los objetos de acuerdo con las características indicadas en la programación.

## 4. FUNDAMENTOS TEÓRICOS

### 4.1. INTELIGENCIA ARTIFICIAL

El objetivo principal de la IA es desarrollar agentes inteligentes que aprendan y evolucionen en entornos desconocidos. Un agente artificial con inteligencia humana en sentido general debe ser capaz de resolver problemas individuales que solucionan un aprendizaje automático en conjunto con otras ramas más especializadas de la inteligencia artificial (Greene, Hoffmann, & Stark, 2019). Por lo tanto, necesita una combinación compleja de tecnologías. En general, la inteligencia artificial (IA) incluye muchos subcampos que comprenden y abstraen aspectos de las habilidades humanas que podrían describirse como inteligencia y tratar de lograrlo en las máquinas.

### 4.2. VISIÓN ARTIFICIAL

Los modelos 3D parciales del entorno ahora se pueden calcular con precisión a partir de miles de fotografías parcialmente superpuestas utilizando métodos confiables. Dado un conjunto de vistas lo suficientemente grande para un objeto o elevación determinados.

Utilizar la visión artificial, se puede visualizar las imágenes y tratar de combinar el reconocimiento de formas, tamaño, tipo y color de cajas. Pero a pesar de todos estos avances, el sueño de que las computadoras interpreten imágenes como se puede visualizar en la ilustración 2 al mismo nivel que para un niño sigue siendo difícil de alcanzar.



*Ilustración 2 Brazo robótico seleccionando objetos por color (Cueto, 2022)*

### **4.3. MACHINE LEARNING**

Machine Learning es una parte de la inteligencia artificial que debe su trabajo al trabajo de algoritmos que procesan datos que las máquinas reciben constantemente. Por lo tanto, se puede estar seguros de que el sistema puede aprender sin programar. Así podrás identificar patrones y sacar tus propias conclusiones. (Serrano, 2021).

### **4.4. MACHINE LEARNING E INDUSTRIA 4.0**

Machine Learning tiene ventajas en la industria como un recurso útil en casi toda la cadena de suministro. El uso de estas herramientas es uno de los elementos fundamentales de la Industria 4.0. Es una tecnología que combina algoritmos de aprendizaje automático con imágenes capturadas por una cámara. (Mecalux, 2020)

Entre las ventajas que obtendrían las empresas al implantar sería:

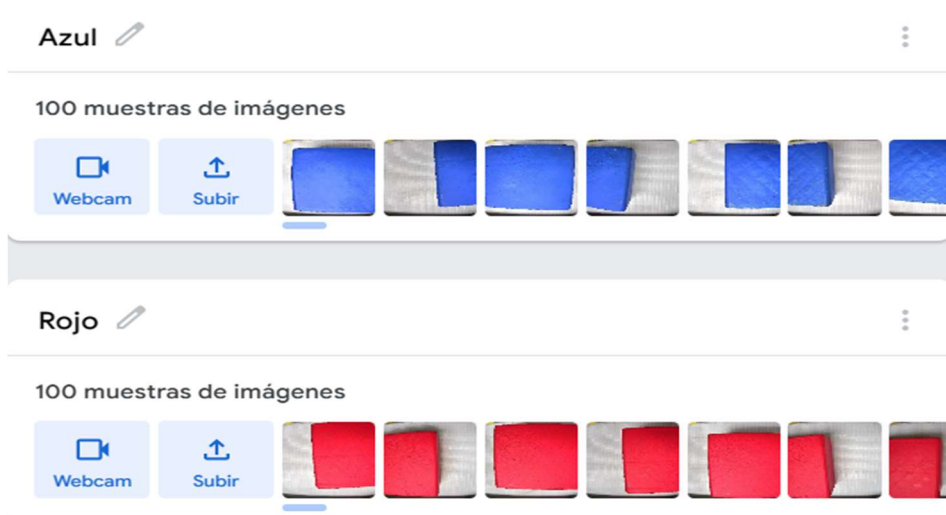
- Planificación de alto nivel, lo que también permite que los recursos humanos realicen tareas de creación de valor de manera más eficiente.
- Facilitan la automatización de procesos, lo que permite que los recursos humanos realice estén más disponibles para realizar tareas.

### **4.5. TEACHABLE MACHINE.**

Es una herramienta desarrollada por Google para crear modelos de aprendizaje automático para reconocer imágenes, sonidos y posturas que está basada en web de una forma más sencilla y accesible para cualquier persona de cualquier edad. (Teachable Machine , conceptos básicos)

Para el proyecto de imágenes se puede obtener la información mediante imágenes por la webcam de la computadora o por una cámara externa. Una vez recopilada las imágenes, se prepara el modelo y estas a la vez se puede exportar para la ejecución de proyecto aplicativos como se muestra en la ilustración 3.





*Ilustración 3 Proceso de creación del modelo con la herramienta*

En esta aplicación se puede aplicar algunos diversos de lenguajes como Python, Glitch, P5.js, Node.js entre otra que son compatibles en el sitio de JavaScript. Además, pueden ser utilizadas en otras plataformas entre una de estas que es Arduino y en el formato que se está utilizando en este proyecto. ([TEKIOS, Octubre 14,2022](#))

#### **4.5.1. VALORES PREDETERMINADOS PARA LA PREPARACIÓN DEL MODELO EN LA HERRAMIENTA DE TEACHABLE MACHINE.**

##### **4.5.1.1. ÉPOCAS**

Este dato hace que el programa se ejecute n veces de procesamiento a todo el conjunto de datos de preparación. Según la herramienta de Teachable Machine, los datos de preparación están basada en el 85 % de las muestras de las clases como se muestra en la ilustración 4. Mientras se escoja el valor mayor, este modelo será mejor en ofrecer resultados predictivos.

Avanzado

Épocas: 50

*Ilustración 4 Valor de época*

#### 4.5.1.2. MUESTRAS DE PREPARACIÓN

Son datos usados o perseguidos para preparar el modelo y la clasificación de muestras en función de las clases definida previamente pero solo se toma el 85% de las muestras totales en una clase.

#### 4.5.1.3. TAMAÑO DE LOTE

Es el conjunto de muestras usadas en la iteración de la preparación del modelo. Según la herramienta de Teachable Machine el tamaño del lote, solo tiene 6 opciones para escoger clases como se muestra en la ilustración 5. Generalmente, este valor no debería cambiar o ajustar para obtener buenos resultados cuando se complete una época. Por ejemplo, si se tiene 160 imágenes y se elige el tamaño de lote de 32, significa que las 160 imágenes divididas por 32, da por resultado 5 lotes. Cuando sea procesado los 5 lotes, se puede indicar que se ha completado una época.



*Ilustración 5 Tamaño de lote*

#### 4.5.1.4. TAZA DE APRENDIZAJE.

Es un parámetro más importante y crítico que determina la rapidez que durante el entrenamiento de un proyecto. Controla el tamaño de actualizaciones que se origina durante el entrenamiento. se ajusta parámetros de un modelo o proyecto. Generalmente, viene definido con un valor muy pequeño y es preferible dejar con el valor indicado para mantener el equilibrio adecuado como se muestra en la ilustración 6, debido a que puede influenciar mayormente en la eficacia del aprendizaje del modelo.

## Tasa de aprendizaje:

0,001 

*Ilustración 6 Tasa de aprendizaje en la preparación del modelo*

### 4.5.2. GRÁFICA DE NIVEL DE RENDIMIENTO DE MODELO EN LA HERRAMIENTA DE TEACHABLE MACHINE

A pesar de que muy pocas personas no ven esta sección, se puede analizar el nivel de rendimiento del proyecto por medio de la precisión de clase, la matriz de confusiones, precisión de época y pérdida de época.

#### 4.5.2.1. PRECISIÓN POR CLASE

Según la herramienta de Teachable Machine, esta precisión está basada en el 15% de las muestras que no fueron usadas en la preparación del modelo. La misma que se utiliza para realizar la comprobación de rendimiento del modelo con datos nuevos como se muestran en la ilustración 7, en la que se puede ver en que el modelo confunde o no con las clases planteadas.

#### Precisión por clase

CLASS	ACCURACY	# SAMPLES
Azul	1.00	15
Rojo	1.00	15
Verde	1.00	15
Banda	1.00	15
Otro	1.00	15

*Ilustración 7 Precisión por clase*

#### 4.5.2.2. MUESTRAS DE PRUEBA

Son datos usados o no perseguidas para comprobar el rendimiento del modelo. Estos datos son diferentes a las muestras de preparación donde se toma el 15% de las muestras restantes en una clase.

#### 4.5.2.3. MATRIZ DE CONFUSIÓN

Herramienta que permite ver el desempeño para conocer y evaluar el algoritmo o proyecto de aprendizaje. En la matriz se puede ver el número de predicción de cada clase y las instancias en la clase real. En la matriz se puede ver un valor diagonal que permite evaluar con mayor exactitud la capacidad predictiva y rendimiento del modelo.

En la ilustración 8, se indica el grado de precisión con respecto a cada una de las clases en un proyecto e indica que comparten características similares, lo que quiere decir que el modelo es capaz de generalizar y realizar predicciones muy precisas.

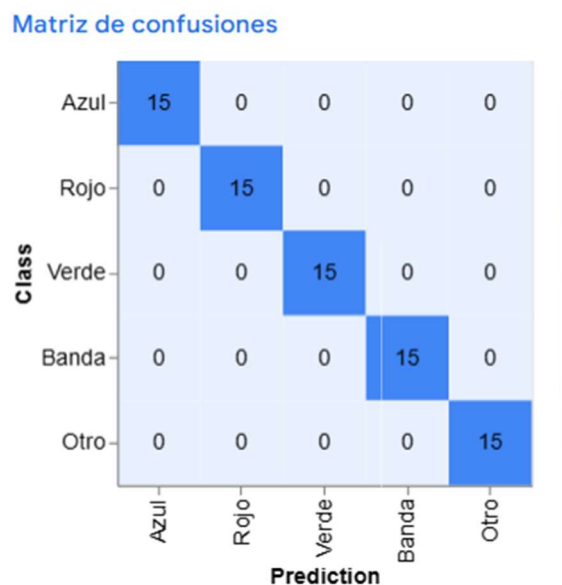


Ilustración 8 Matriz de confusión

#### 4.5.2.4. PRECISIÓN DE ÉPOCA

Es un indicador de precisión del entramiento del modelo por época y según una breve variación en los valores predeterminado se puede indicar que la predicción es perfecta cuando el modelo es igual a 1. En esta ilustración 9, se puede ver que la línea naranja tiende a 1 desde que se completa una época por lo que se puede indicar que el modelo es tan preciso en la fase de entrenamiento del modelo debido a la misma cantidad del tamaño muestras por clases.

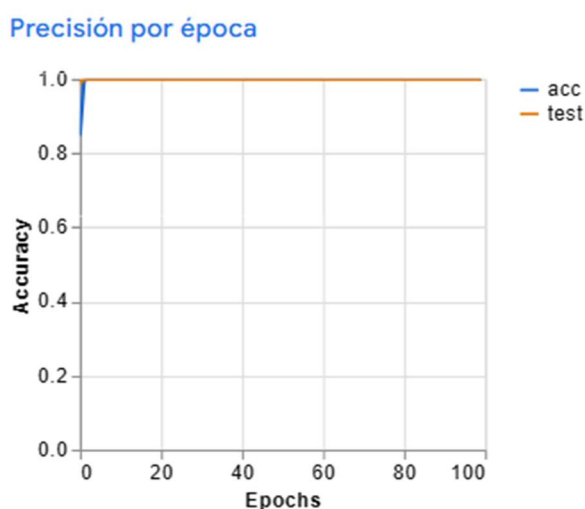


Ilustración 9 Precisión de época

#### 4.5.2.5. PÉRDIDA DE ÉPOCA

Mide el nivel de aprendizaje o nivel confianza en este modelo en lo que se puede ver que es casi perfecta debido a que la curva de color naranja tiende a 0 como se puede visualizar en la ilustración 10.

Si el valor en alto, el modelo puede generar gráficas incompresibles o confusas. Si el a es demasiado bajo, el modelo puede perder información valiosa en el momento del entrenamiento.

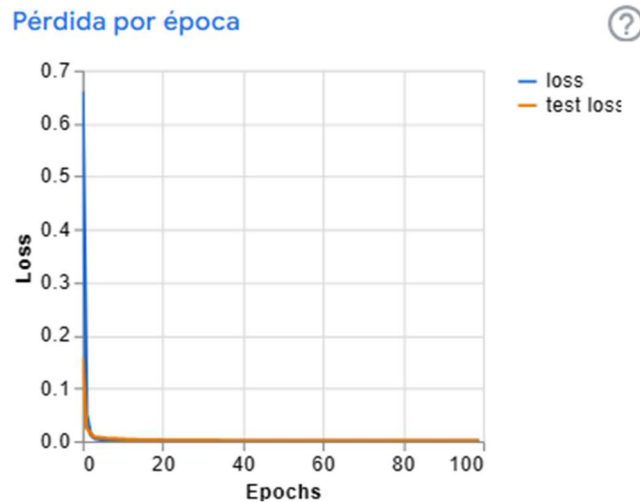


Ilustración 10 Precisión por clase y Pérdida por clase

### 4.5.3. EXPORTAR MODELO

El modelo creado con Teachable Machine, son modelos en Tensor Flow.js que puede utilizarse en cualquier sitio que utilice JavaScript y que son compatibles con las herramientas de Glitch, P5.js entre otras. ([Teachable Machine, FAQ, s.f.](#)). Estos modelos pueden ser usados en otras plataformas como Coral y Arduino que será utilizado en el desarrollo e implementación de un modelo de Machine Learning de Google para la clasificación de objetos por color. Para exportar el modelo se debe escoger el icono de **exportar modelo** como se muestra en la ilustración 11.

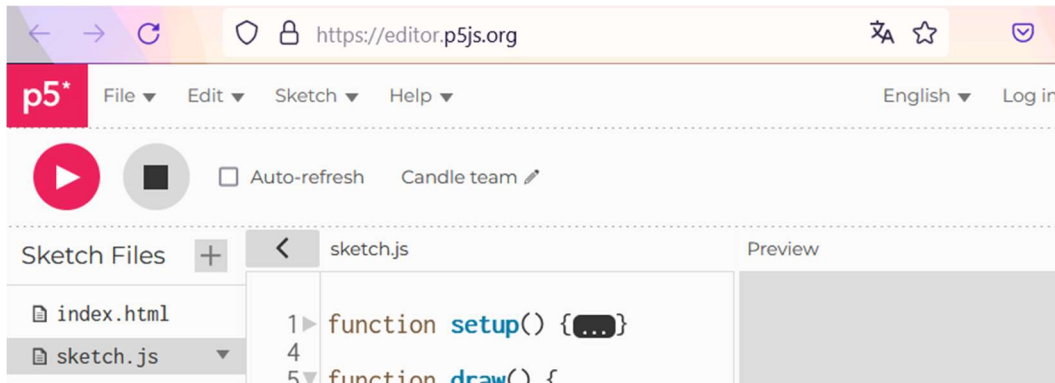


Ilustración 11 Exportar modelo desde Teachable Machine

### 4.6. P5.JS WEB EDITOR

Es una biblioteca de JavaScript de código abierto que permite hacer páginas web creativos. ([P5.js, s.f.](#)). Fue creado por Lauren McCarthy y está dirigido por Quinlan Ye. Este proyecto forma parte de Processing Foundation que apoya a una comunidad de colaboradores, la misma que crea software y materiales educativos. Estas librerías son muy sencillas y accesibles para diseñadores, educadores y principiantes que buscan crear piezas

de artes, prototipos y animaciones en el navegador. La herramienta es una web editable como se muestra en la ilustración 12.



*Ilustración 12 Herramienta P5.js*

## 5. MARCO METODOLÓGICO

La metodología que se utilizó para clasificar colores fue de revisar las características técnicas, facilidad de uso compatibilidad y costo. La selección del hardware, el tipo de cámara e implementación, proceso del prototipo en que se desarrollará este proyecto.

Primero se debe capturar los fotogramas de colores de las cajas y enviarlos a la herramienta Teachable Machine, después obtener el modelo de aprendizaje automático para cargar a un computador y finalmente, clasificar los objetos de acuerdo con las características indicadas en la programación de la Web editable de J5.js.

### 5.1. PASOS PARA HACER UN PROYECTO DE IMAGEN EN LA HERRAMIENTA TEACHINE MACHINE EN GOOGLE

Al visualizar la página, se indican un breve y rápido concepto de esta herramienta. También se indica como se utiliza, el paso a paso a seguir para poder recopilar las imágenes, sonido y posturas.

Además, se facilitan tutoriales para revisar y poder usar esta herramienta.

### 5.1.1. INICIAR SESIÓN EN TEACHABLE MACHINE

Primero abrir el link: <https://teachablemachine.withgoogle.com/>

Después se debe iniciar sesión desde una cuenta de Gmail.

Finalmente, seleccionar **Abrir proyecto**, seguido se selecciona **Nuevo proyecto de imagen** y se elige **Modelo de imagen estándar** como se puede mostrar en la ilustración 13.



Ilustración 13 Ingreso a la página de Teachable Machine

Una vez seleccionada, se puede visualizar un cuadro como ventana con la palabra **clases** y poder editar con nombres donde se colocarán las muestras. Además, se pueden agregar n clases. Finalmente, se puede ver como añadir las imágenes a trabajar como visualizar en la ilustración 14.



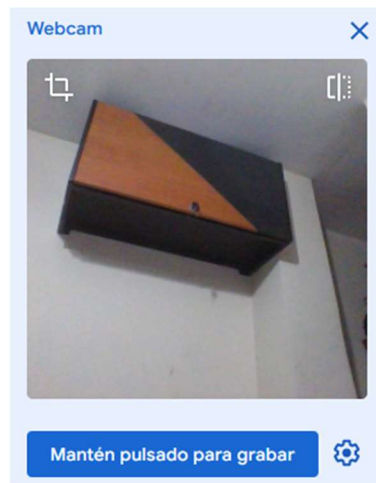
Ilustración 14 Clases de muestras



### 5.1.2. OBTENCIÓN DE LOS FOTOGRAMAS O IMÁGENES PARA LAS CLASES

Después de colocar nombres a las clases que se va a trabajar, se procede a obtener las imágenes para cada clase e ir guardando el proyecto. Las imágenes se las puede obtener por 2 formas:

- a) Por Webcam del ordenador o cámara externa como se muestra en la ilustración 15.



*Ilustración 15 Obtención de imágenes por medio de la cámara del computador*

- b) Por archivos desde el ordenador, en la que puede cargar en la herramienta como importar imágenes desde el Google drive como se muestran en la ilustración 16.



*Ilustración 16 Visualización para obtener imágenes por medio de archivos*

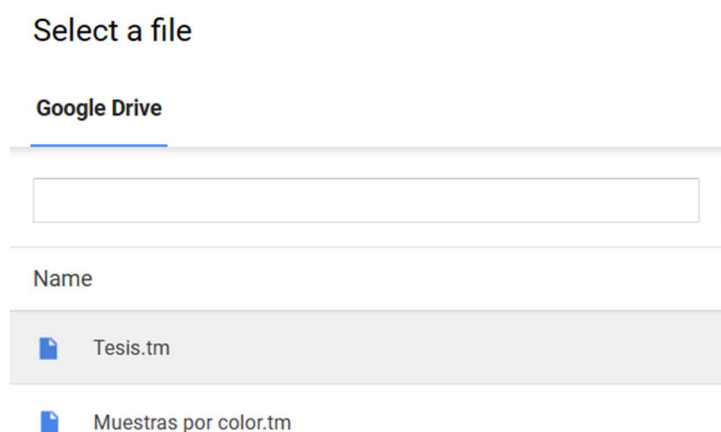
### 5.1.3. GUARDAR IMÁGENES EN EL PROYECTO

Después de obtener las imágenes, se necesita guardar o asignar un nombre al proyecto en tu drive como se indica en la ilustración 17.



*Ilustración 17 Guardar imagen o proyecto*

Además, al guardar el proyecto permitirá volver a restaurar para volver a usarlo como se muestra en la ilustración 18.



*Ilustración 18 Proyectos en Google Drive*

Cada vez que se realiza un cambio o se agrega fotos, se debe guardar de tal forma que el proyecto queda actualizado, caso contrario se puede perder información cuando se desea obtener el modelo a entrenar con las muestras.

### 5.1.4. PASOS PARA OBTENER EL MODELO

Para obtener el modelo hay que considerar algunas variables que se puede visualizar en la pestaña de avanzado como valores predeterminados entre ellas el valor de época, tamaño de lote, y la taza de aprendizaje.

Se tienen 5 clases y se toman 100 imágenes por cada clase dando un total de 500 imágenes, por lo que el patrón para que el programa se ejecute el procesamiento a todo el conjunto de datos de preparación que se ha escogido es 100 épocas.

A este conjunto de muestras usadas en la iteración de la preparación del modelo se ha escogido en número de lote de 16 muestras, lo que significa que las 500 imágenes dividida por 16 da como resultado 32 lotes, que cuando sean procesado los 32 lotes, se puede indicar que se ha completado una época. Generalmente, este valor no debería cambiar o ajustarse para obtener buenos resultados cuando se complete una época.

Para el valor de la tasa de aprendizaje que viene definido es preferible mantener en 0.001, debido a que puede influenciar mayormente en la eficacia del aprendizaje del modelo.

Al obtener el modelo por medio de la herramienta, se puede ver en el icono de **Más datos** en donde se abre una ventana extra que muestra de cómo evalúa el entrenamiento de modelo en Teachable Machine. En las cuales se puede ver que tan preciso fue la precisión por clase. El usuario puede monitorear los valores predeterminados del modelo de cada muestra evaluada de cada clase que esta precisión está basada en el 15 % de las muestras que fueron usadas en la preparación del modelo, donde se muestra que el modelo no confunde las 15 imágenes de las clases planteadas en el modelo, lo que significa que es un modelo casi perfecto, como se muestra en la ilustración 19.

CLASS	ACCURACY	# SAMPLES
Azul	1.00	15
Rojo	1.00	15
Verde	1.00	15
Banda	1.00	15
Otro	1.00	15

*Ilustración 19 Precisión por clases*

También, el usuario puede validar el desempeño del entrenamiento de las muestras con la matriz de confusión, en la que puede ver los indicadores como la cantidad de predicciones correctas que tiene su predicción con respecto a cada una de las clases como se muestra en la ilustración 20.

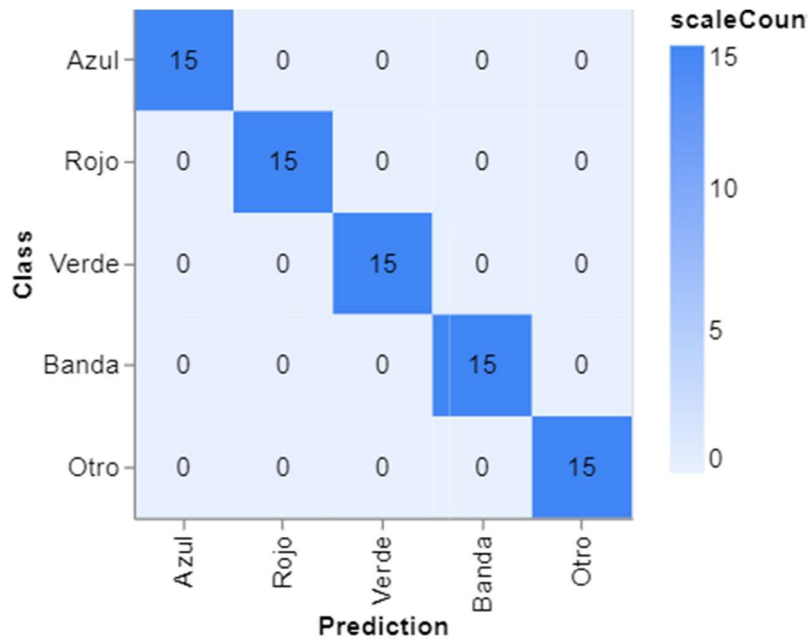
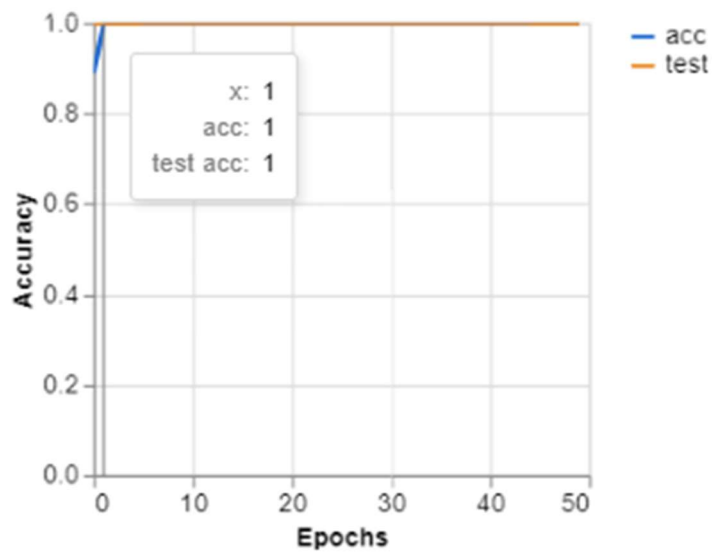


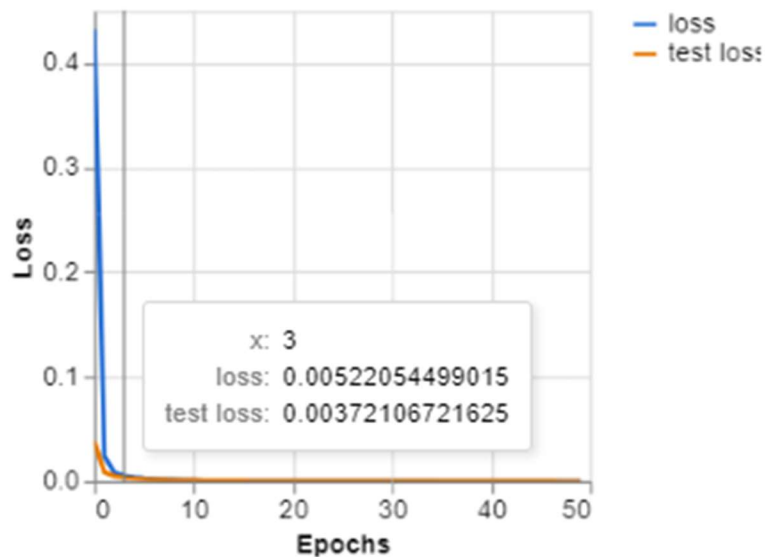
Ilustración 20 Matriz de confusión

La herramienta utiliza el 15 % de las muestras para realizar pruebas de rendimiento del modelo, por lo que permite ver la precisión del entrenamiento del modelo por época y según una breve variación en los valores predeterminado se puede indicar que la curva naranja es casi lineal desde el inicio, por lo que la predicción es casi perfecta debido a que tiende a 1 como se puede visualizar en la ilustración 21.



*Ilustración 21 Precisión por época*

Finalmente, se puede ver las pérdidas por épocas de tal forma que calcula el nivel de aprendizaje al predecir la clasificación de las imágenes, el margen de pérdida es de 0.005 en lo que se puede ver que es casi perfecta debido a que la curva naranja tiende a 0 como se puede visualizar en la ilustración 22.



*Ilustración 22 Pérdida por época*

En la herramienta Teachable Machine se puede reconocer el objeto por color y entre algunas pruebas, se puede ver los resultados como se indica lo siguiente:

### Reconocimiento de la clase Otro.

Se puede ver que reconoce al 100% el objeto que pertenece a la clase Otro, como se muestra en la ilustración 23.



*Ilustración 23 Reconocimiento de la clase Otro en Teachable Machine por cámara del ordenador*

### Reconocimiento de la clase Azul

En este se puede ver que reconoce al 100% el objeto que pertenece a la clase Azul, como se muestra en la ilustración 24.



Ilustración 24 Reconocimiento de la clase Azul en Teachable Machine por cámara del ordenador

### Reconocimiento de la clase Rojo

Se puede ver que reconoce al 100% el objeto que pertenece a la clase Rojo, como se muestra en la ilustración 25.

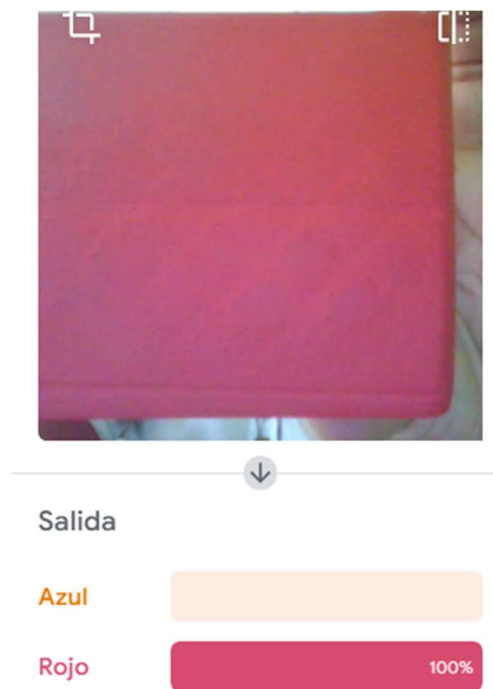
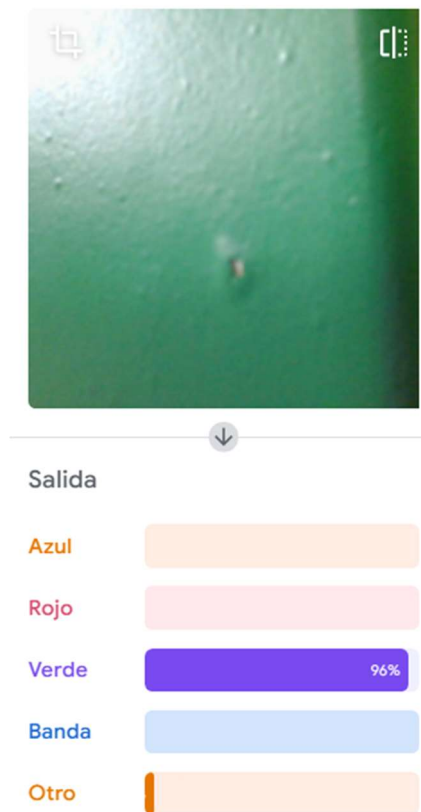


Ilustración 25 Reconocimiento de la clase Rojo en Teachable Machine por cámara del ordenador

## Reconocimiento de la clase Verde

Se puede ver que reconoce al 96% el objeto que pertenece a la clase Verde, pero se pierde 4% debido a la luminosidad que le llega al objeto, como se muestra en la ilustración 26.



*Ilustración 26 Reconocimiento de la clase Verde en Teachable Machine por cámara del ordenador*

### 5.1.4.1. PASOS PARA EXPORTAR MODELOS

Para exportar el modelo, se deben escoger el botón que dice **Exportar modelo**, como se muestra en la ilustración 27.



*Ilustración 27 Botón de exportar modelo*



Inmediatamente, escoger el ícono que dice **subir modelo (enlace para compartir)** como se muestra en la ilustración 28, para poder generar el URL del modelo.

Exporta tu modelo:



*Ilustración 28 Subir modelo*

Despues, se actualiza el modelo entrenado y aparece el URL del modelo como se muestra en la ilustración 29, en la que se muestra sombreada en celeste para poder ser utilizado en otras herramientas de programacion web.

Exporta tu modelo:



Tu enlace para compartir:



*Ilustración 29 Modelo en nube y enlace para compartir*

Finalmente, se copia el enlace. En la misma se puede ver que el URL fue copiado modelo como se muestra en la ilustración 30 y listo para ser anidado en otra herramienta en que se vaya a trabajar con este modelo entrenado.

Tu enlace para compartir:



Cuando subas el modelo, Teachable Machine lo alojará en este enlace. (Preguntas frecuentes: [¿Quién puede usar mi modelo?](#))

*Ilustración 30 Copiado de URL de modelo*

### 5.1.5. CLASIFICACIÓN DE OBJETOS POR COLOR POR MEDIO DE LA HERRAMIENTA P5.js WEB EDITOR.

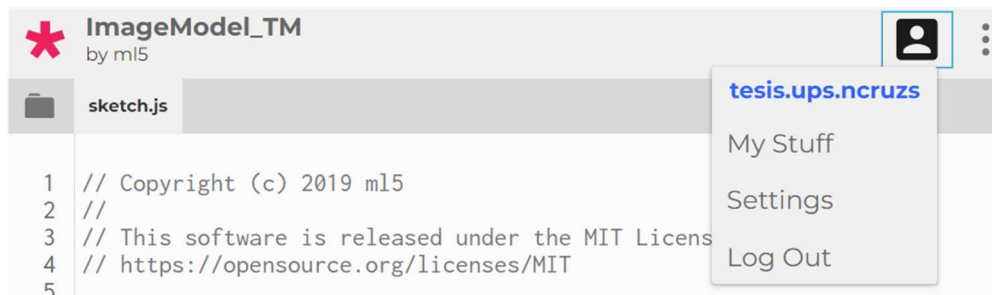
Como se tiene iniciada la sesión de Gmail se puede ingresar al web editor P5.js. desde la herramienta de Teachable Machine, y en donde dice **fragmentos de código para usar el modelo** se debe escoger **p5.js Web Editor** como se muestra en la ilustración 31.

Fragmentos de código para usar el modelo:



*Ilustración 31 Fragmentos de código para usar modelo.*

Una vez seleccionada, se abre una ventana externa, donde se puede ver un ejemplo de código de programa, en la cual se puede editar como se muestra en la ilustración 32.



*Ilustración 32 Herramienta de web editable de P5.js*

Después se debe guardar el proyecto con un nombre para reconocerlo. Además, puedes seleccionar en **Auto-refresh** para que el proyecto se actualice cada vez que se realiza algún cambio en el código en la web, como se muestra en la ilustración 33.



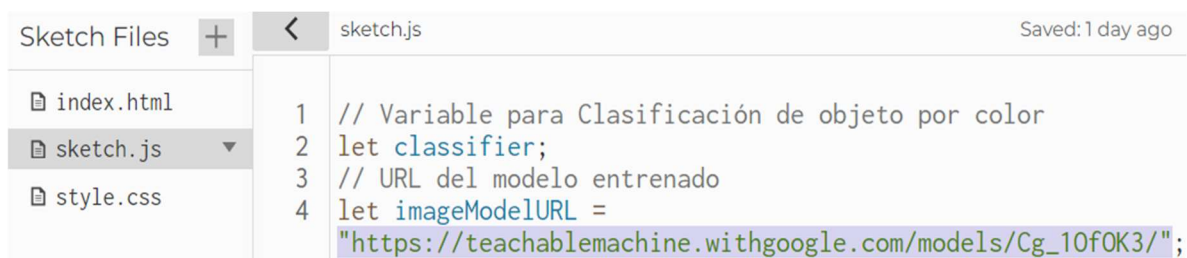
*Ilustración 33 Autoguardar proyecto en P5.js.*

Con respecto a la programación, se puede editar el código del index.html donde se coloca el título del proyecto e información general como se muestra en la ilustración 34. El código completo se encuentra disponible al final de este informe en anexos.

```
17 <body>
18   <h1>CLASIFICACIÓN DE OBJETOS POR
    COLOR</h1>
19   <p>Este es una demostración de
    clasificación de imágenes utilizando un
    modelo con Teachable Machine de Google.
20
21   <p>Proyecto de tesis: " DESARROLLO E
    IMPLEMENTACIÓN DE UN MODELO DE MACHINE
    LEARNING DE GOOGLE PARA LA CLASIFICACIÓN DE
    OBJETOS POR COLOR " </p>
22   <script src="sketch.js"></script>
23 </body>
```

Ilustración 34 Index de la herramienta de P5.js

En el Sketch.js se puede editar el código y lo que va a realizar el proyecto. Además. Al iniciar el código, se designa la variable **classifier**, la que se encarga de clasificar. Se debe ingresar el URL del modelo entrenado en Teachable Machine en la línea 4 donde se designa la variable **imageModelURL** la misma que se ejecutará cada vez necesite reconocer el color de un objeto tal como se muestra en la ilustración 35.



```
Sketch Files + < sketch.js Saved: 1 day ago
index.html
sketch.js
style.css
1 // Variable para Clasificación de objeto por color
2 let classifier;
3 // URL del modelo entrenado
4 let imageModelURL =
  "https://teachablemachine.withgoogle.com/models/Cg_10f0K3/";
```

Ilustración 35 Link del modelo en la herramienta de P5.js

Para ver el video y almacenar la clasificación, se designa a las variables para generar el video, generar el video a la inversa y para almacenar la primera clasificación como se muestra en la ilustración 36.

```

6 // Para Video
7 let video;
8 let flippedVideo;
9 // Almacenar la clasificación
10 let label = "";

```

*Ilustración 36 Designa etiqueta de video y almacenar*

Para iniciar código, en la línea 13 se solicita la función **preload** como se muestra en la ilustración 37. La misma que se encarga de cargar el modelo con las imágenes entrenadas.

```

12 // Carga del primer modelo
13 function preload() {
14     classifier = ml5.imageClassifier(imageModelURL +
15     "model.json");

```

*Ilustración 37 Función de precarga*

Después, se solicita la función **setup**, en donde se declaran otras variables y la primera función a ejecutar el código como se muestra en la ilustración 38. En la línea 18, se declara la variable para crear el fondo de pantalla con su rango y alto de la cámara. En la línea 22, se declara la variable para capturar el video a través de la webcam. En la línea 23, se indica la medida que tendrá la ventana del video y en la línea 24, se ejecuta para esconder el video Después, en la línea 26, se designa la variable de predicción del modelo entrenado que se va a clasificar de la donde se conecta al video entrenado que la función que se encarga de transferir el contenido de video en tiempo real y finalmente se llama a la variable para seguir el ciclo del proceso de clasificación del video de acuerdo al modelo entrenado cada vez que se la cámara recepta una muestras y la muestra en la herramienta.

```

17 function setup() {
18   createCanvas(320, 260);
19
20
21   // Crea el video
22   video = createCapture(VIDEO);
23   video.size(320, 240);
24   video.hide();
25
26   flippedVideo = ml5.flipImage(video);
27   // Inicia la clasificación
28   classifyVideo();
29 }

```

*Ilustración 38 Inicio del código: Función setup*

Para poder ejecutar contantemente, se solicita la función **draw** para que se ejecute cada vez que un objeto es vista por la webcam, como se muestra en la ilustración 39. Dentro del bucle se deben declarar las siguientes de variables: en la línea 32, puede ver que está declarada la variable para crear el fondo de pantalla; en la 34 se carga el video con su respectiva posición en x y ya la reversa; de la línea 37 a la 40, se declara la etiqueta de resultados con su respectivo tamaño, color, dirección.

```

31 function draw() {
32   background(0);
33   // Dibuja el video
34   image(flippedVideo, 0, 0);
35
36   // Crea la etiqueta
37   fill(255);
38   textSize(16);
39   textAlign(CENTER);
40   text(label, width / 2, height - 4);
41 }

```

*Ilustración 39 Función draw*

Se declara la variable para la clasificación como se muestra en la ilustración 40. Desde la línea 44 a la 47, se muestra para ejecutar el video en reversa y la clasificación que de ejecuta de acuerdo al modelo entrenado e indicando el resultado. Además, se remueve el resultado de acuerdo a lo que se muestre desde la webcam.

```

43 // Se obtiene el cuadro de pantalla del video
44 function classifyVideo() {
45     flippedVideo = ml5.flipImage(video);
46     classifier.classify(flippedVideo, gotResult);
47     flippedVideo.remove();
48 }

```

*Ilustración 40 Función de clasificación*

Para obtener resultado, en la línea 51 se declara la función de obtener resultado como se muestra en la ilustración 41. Dentro de este ciclo, en la línea 32 se inicia una comparación de error donde se mostrará en la consola caso contrario retorna a seguir a clasificación de acuerdo al modelo entrenado. En la línea 59 se muestra la componente del texto que no se puede editar y en la línea 63 indica que puede seguir realizando el bucle de clasificar por medio de la webcam.

```

50 // Obtención del resultado
51 function gotResult(error, results) {
52     // En caso de error
53     if (error) {
54         console.error(error);
55         return;
56     }
57     // Los resultados se muestran en la matriz
ordenada de confianza.
58     // console.log(results[0]);
59     label = results[0].label;
60
61
62     // Clasifica nuevamente!
63     classifyVideo();
64 }

```

*Ilustración 41 Obtener resultado y error*

El código completo se encuentra disponible al final de este informe en anexos.

Al finalizar el código, debe hacer click en botón rojo de play como se muestra en la ilustración 42, para poder realizar el bosquejo del código editado del sketch.js para realizar el programa para mostrar por medio de la cámara los resultados de la clasificación de objetos por color e indicar en la etiqueta por medio de una pantalla.

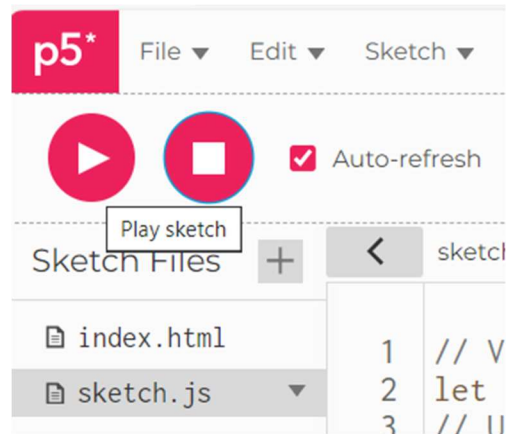


Ilustración 42 Index de la herramienta de P5.js

## 6. RESULTADOS

Se obtuvo las capturas de las imágenes o fotogramas desde la Webcam y desde la cámara del ordenador para la aplicación de Teachable Machine como se puede ver en la ilustración 43. Además, se muestra cuando se ejecutaba la captura de los fotogramas en la herramienta de Teachable Machine de Google.

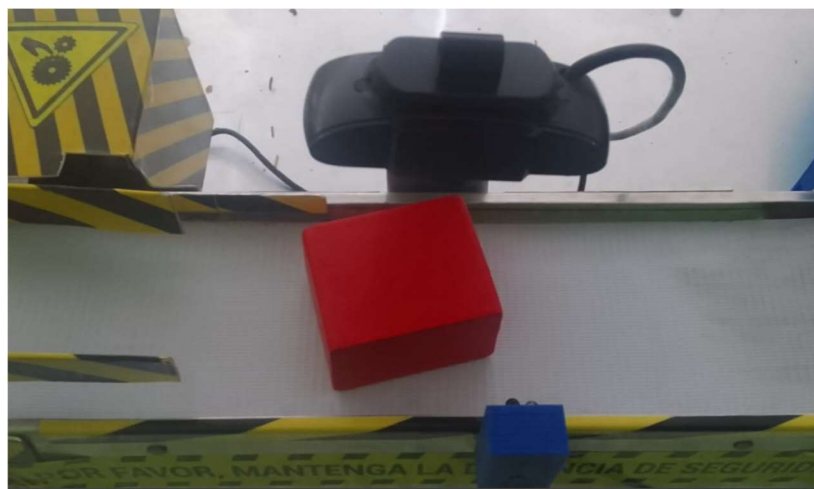


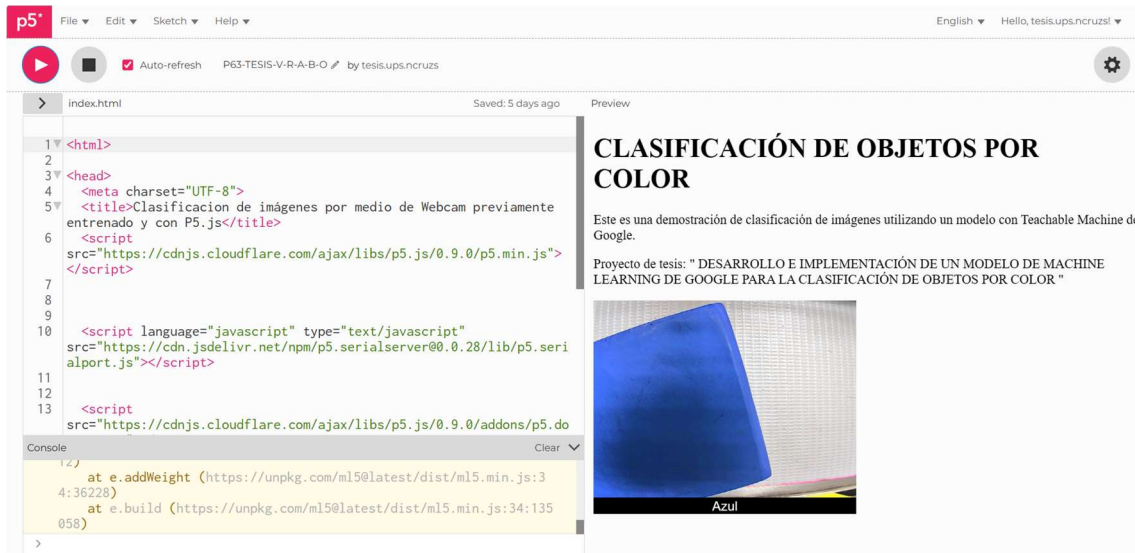
Ilustración 43 Prototipo con Webcam apagada



Se obtuvo el URL del modelo de aprendizaje automático que es:

[https://teachablemachine.withgoogle.com/models/Cg\\_1OfOK3/](https://teachablemachine.withgoogle.com/models/Cg_1OfOK3/)

Al cargarlo al computador se puede ver como ejecuta la programación de acuerdo al modelo entrenado como se muestra en la ilustración 44.

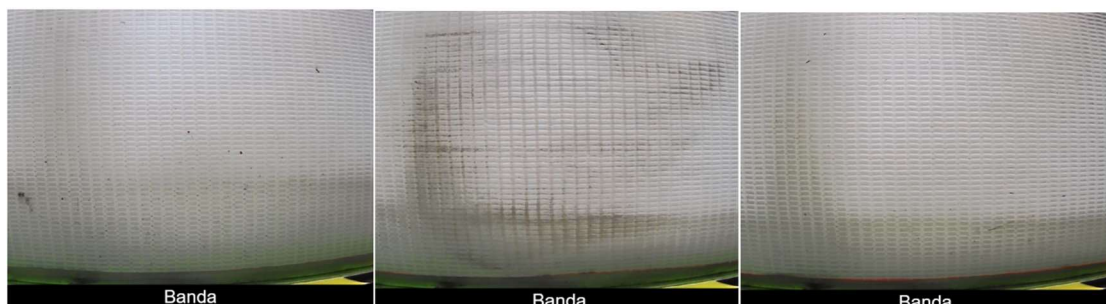


*Ilustración 44 URL en la herramienta P5.js*

Se hizo diferentes pruebas realizadas desde el ordenador como desde la Webcam por medio desde Teachable Machine como desde las herramientas de P5.js en las cuales ambas pruebas dando como éxito el reconocimiento de los objetos por color y previamente con el modelo entrenado, dando como resultado lo siguiente:

### **Pruebas para la clase Banda.**

Se puede ver que en diferentes posiciones o rodamiento de la banda es reconocida como la clase Banda, como se muestra en la ilustración 45.

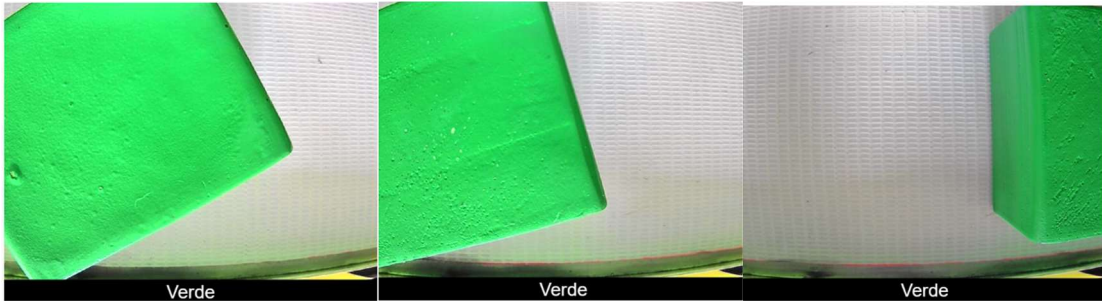


*Ilustración 45 Reconocimiento de la clase Bandal en P5.js por Webcam*



### Pruebas para la clase objeto Verde.

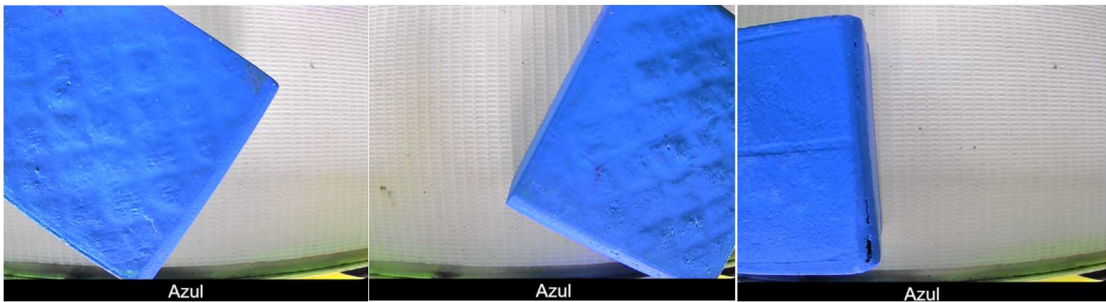
Se puede ver que en diferentes posiciones es reconocida como la clase Verde, como se muestra en la ilustración 46.



*Ilustración 46 Reconocimiento de la clase Verde en P5.js por Webcam*

### Pruebas para la clase objeto Azul.

Se puede ver que en diferentes posiciones es reconocida como la clase Azul, como se muestra en la ilustración 47.



*Ilustración 47 Reconocimiento de la clase Azul en P5.js por Webcam*

### Pruebas para la clase objeto Rojo.

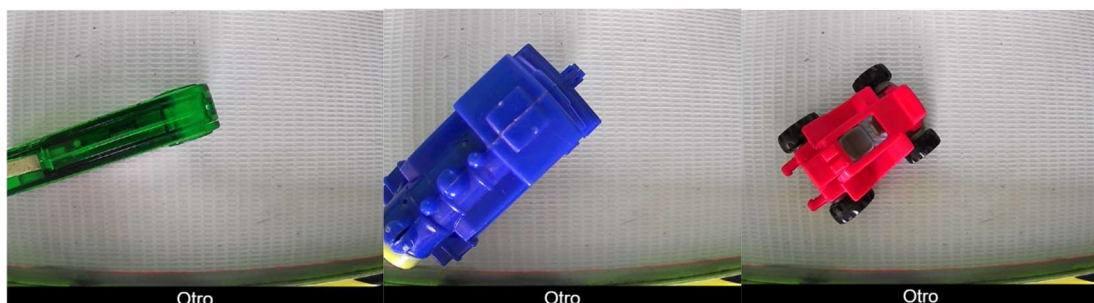
Se puede ver que en diferentes posiciones es reconocida como la clase Rojo, como se muestra en la ilustración 48.



*Ilustración 48 Reconocimiento de la clase Rojo en P5.js por Webcam*

### Pruebas para la clase Otro.

Se puede ver que en diferentes posiciones es reconocida como la clase Otro, como se muestra en la ilustración 49.



*Ilustración 49 Reconocimiento de la clase Otro en P5.js por Webcam*

Se desarrolló la programación en el software P5j Editor como código base de intercomunicación con la aplicación de Teachable Machine para el proceso de reconocimiento de objetos por color, mediante una programación de fácil manejo para los usuarios, con el objetivo de cumplir con el funcionamiento de ubicar las cajas de colores en que se considera que el entrenamiento fue exitoso.

Este proyecto se encuentra disponible en una cuenta de Gmail donde están los archivos de imágenes junto con la programación para que pueda ser ejecutada y cambiada para futuros proyectos o a fines.

## 7. CRONOGRAMA

El plan de ejecución del proyecto se lo realizó en un lapso de tres meses que comprenden desde el mes de octubre del 2023 hasta el mes de enero del 2024, las actividades realizadas se muestran en la tabla 1 de la siguiente manera:

**Tabla 1** Cronograma de actividades

No.	Tareas para realizar durante el proceso de elaboración del artículo científico	octubre-23				nov-23				dic-23				ene-23			
		Semanas															
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Revisión con el tutor para cambio tema y objetivos.	■	■														
2	Aprobación anteproyecto.			■	■												
3	Recopilación de fotogramas.			■	■		■										
4	Obtener el modelo para la simulación							■	■								
5	Pruebas y simulación del modelo							■	■								
6	Primera revisión de avances al tutor.									■	■						
7	Investigación sobre las herramientas de Teachable Machine y la aplicación Web Editor P5.js.					■	■			■			■	■	■		
8	Avances del libro							■	■	■	■	■	■	■	■		
9	Segunda revisión de avances al tutor														■	■	
10	Finalización del libro de titulación.														■	■	

## 8. PRESUPUESTO

Para el desarrollo e implementación de un modelo de Machine Learning de Google para la clasificación de objetos por color, se ha considerado el tiempo de inversión del trabajo de ingeniería en el desarrollo investigativo, así como para culminar el proyecto. Para desarrollo del proyecto se utilizó parte de un prototipo y a la vez fue necesario adquirir objetos varios, por lo que se elaboró un presupuesto, cuya información se detalla en la tabla.

Descripción de materiales	Cantidad	Costo Unitario	Costo Total
Tiempo de ingeniería (Horas)	175	\$ 2,81	\$ 492
Insumos para el desarrollo del trabajo.	10	\$ 6,50	\$ 65
<b>TOTAL</b>			<b>\$ 557</b>

*Tabla 2 Tabla de presupuesto*

**Nota:** El prototipo de la banda, los cubos y la cámara fue proporcionada por la Universidad.

## 9. CONCLUSIONES

Mediante la conexión entre el computador y la cámara se capturó los fotogramas de colores de las cajas, y se guardó en la herramienta Teachable Machine de Google obteniéndose los archivos en el drive.

Una vez que los fotogramas fueron subidas a la aplicación, se pudo obtener el modelo de las muestras entrenadas.

Además, los resultados de realizar varias pruebas y error se pudo obtener el modelo de aprendizaje automático cumpliendo con el proceso de clasificación de imágenes por color.

## 10. RECOMENDACIONES

Se recomienda la cantidad suficiente de muestras para cada clase y en diferentes posiciones dentro de las clases. La cantidad de datos para el entrenamiento es un factor a considerar, mientras más imágenes se puede demorar el entrenamiento del modelo, pero puede ser un modelo sencillo de aplicar con menos imágenes.

Balancear cada clase, es decir que cada clase debe tener la misma cantidad de muestras para evitar que el modelo tenga **sesgo** hacia la clase con mayor número de muestras.

Debe considerarse como clase aquellos escenarios en los que no se espera el objeto habitual.

Realizar un preentrenamiento para poder estimar el número de épocas que permite tener un error mínimo aceptable.

## 11. REFERENCIA BRIBLIOGRÀFICA

*Teachable Machine* ([Google, conceptos básicos](#))

*Utilizando Visión Estereoscópica* ([Naranjo, Nono & Banda, 2014](#))

*Colores espectrales* ([David Camilo López, Tarea](#))

*Revisión de la visión artificial 3D y sus aplicaciones en la edad de Industria 4.0*  
([Lanner,2017](#))

*La visión artificial: una revolución en la Industria 4.0* ([UNIR Revista, 2021](#))

*Comprendiendo la importancia de las tasas de aprendizaje en los modelos de aprendizaje automático* ([CIVITAI, Aguilar, JUNIO8,2023](#))

*Detección precoz de sarna en rebecos mediante redes neuronales convolucionales*  
([Martines, Guerola, UNIVERSITAT POLITÈCNICA DE VALÈNCIA , 2022](#))

*Evaluación de rendimiento de un modelo de machine Learning* ([Zevallos, julio 8,2022](#))

## 12. ANEXOS

### 12.1. PROGRAMACIÓN

#### 12.1.1. CÓDIGO DEL INDEX EN P5.js

A continuación, se presenta el código que ejecuta o grafica para mostrar la aplicación por medio de video desde la herramienta P5.js:

```
<html>

<head>
  <meta charset="UTF-8">
  <title>Clasificación de imágenes por medio de Webcam previamente entrenado y con
P5.js</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>

  <script language="javascript" type="text/javascript"
src="https://cdn.jsdelivr.net/npm/p5.serialserver@0.0.28/lib/p5.serialport.js"></script>

  <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"
type="text/javascript"></script>
</head>

<body>
  <h1>CLASIFICACIÓN DE OBJETOS POR COLOR</h1>
  <p>Este es una demostración de clasificación de imágenes utilizando un modelo con
Teachable Machine de Google.

  <p>Proyecto de tesis: " DESARROLLO E IMPLEMENTACIÓN DE UN MODELO
DE MACHINE LEARNING DE GOOGLE PARA LA CLASIFICACIÓN DE OBJETOS
POR COLOR " </p>
  <script src="sketch.js"></script>
</body>

</html>
```

### 12.1.2. CÓDIGO DEL SKTCH EN P5.js

A continuación, se presenta el código del sketch.js en la herramienta P5.js que permite ejecutar la clasificación de objetos por color:

```
// Variable para Clasificación de objeto por color
let classifier;
// URL del modelo entrenado
let imageModelURL = "https://teachablemachine.withgoogle.com/models/Cg_1OfOK3/";

// Para Video
let video;
let flippedVideo;
// Almacenar la clasificación
let lábil = "";

// Carga del primer modelo
function preload() {
  classifier = ml5.imageClassifier(imageModelURL + "model.json");
}

function setup() {
  createCanvas(320, 260);

  // Crea el video
  video = createCapture(VIDEO);
  video.size(320, 240);
  video.hide();

  flippedVideo = ml5.flipImage(video);
  // Inicia la clasificación
  classifyVideo();
}

function draw() {
  background(0);
  // Dibuja el video
  image(flippedVideo, 0, 0);

  // Crea la etiqueta
```

```
fill(255);
textSize(16);
textAlign(CENTER);
text(label, width / 2, height - 4);
}

// Se obtiene el cuadro de pantalla del video
function classifyVideo() {
  flippedVideo = ml5.flipImage(video);
  classifier.classify(flippedVideo, gotResult);
  flippedVideo.remove();
}

// Obtención del resultado
function gotResult(error, results) {
  // En caso de error
  if (error) {
    console.error(error);
    return;
  }
  // Los resultados se muestran en la matriz ordenada de confianza.
  // console.log(results[0]);
  label = results[0].label;

  // Clasifica nuevamente!
  classifyVideo();
}
```