

# Sistema de detección de emociones faciales mediante técnicas de Machine Learning adaptado a ROS para un robot de bajo coste basado en Raspberry Pi

Javier Martínez Madruga

[j.martinezma.2018@alumnos.urjc.es](mailto:j.martinezma.2018@alumnos.urjc.es)



Trabajo fin de grado

28 de Junio de 2022



(CC) Javier Martínez

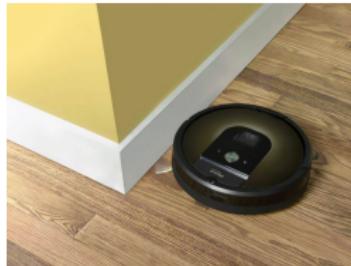
*Este trabajo se entrega bajo licencia CC BY-NC-SA. Usted es libre de (a) compartir: copiar y redistribuir el material en cualquier medio o formato; y (b) adaptar: remezclar, transformar y crear a partir del material. El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia.*

# Contenidos

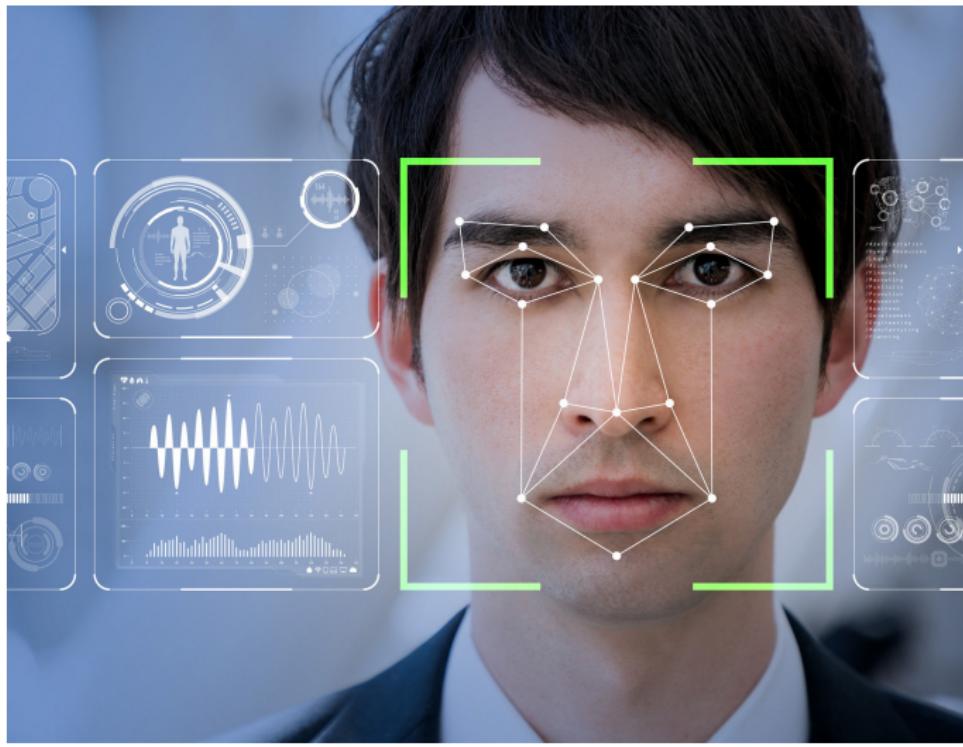
- 1 Introducción
- 2 Objetivos
- 3 Plataforma de desarrollo
- 4 Sistema de detección de emociones
- 5 Estudios de optimización
- 6 Conclusiones

# *Introducción*

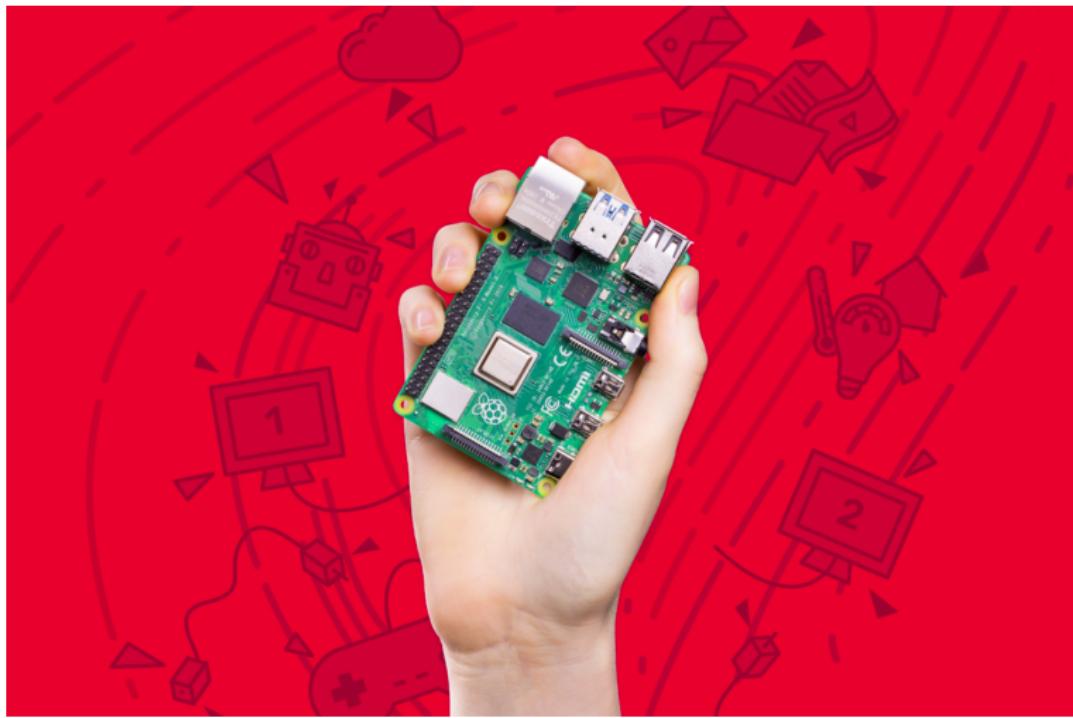
# Robótica de Servicio y HRI



# Visión Artificial y Machine Learning



# Sistemas empotrados



# *Objetivos*

# Descripción del problema

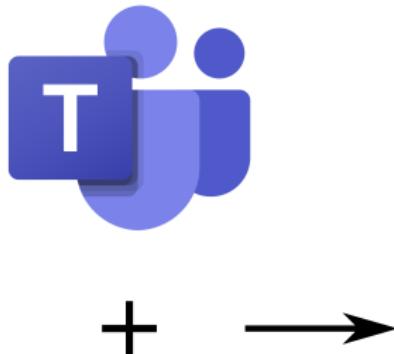
## Objetivo principal

Desarrollar una herramienta de **reconocimiento de emociones faciales** capaz de funcionar en un sistema robótico de **bajo coste**.

## Subobjetivos

- Estudiar el estado del arte.
- Optimizar y adaptar la técnica escogida.
- Generar un dataset de valor.
- Entrenar un modelo con varios clasificadores.
- Integrar la herramienta en ROS.

# Metodología



Screenshot of a GitHub repository page for `jmrtzma / emotion_detection_ros`. The repository is marked as "Public". The page includes a file tree, a detailed description of the project ("Real Time Emotion Detection for Low Cost Robot in ROS"), sections for Overview, Model configuration, and Face examples, and two small video frames demonstrating the emotion detection software running on a Raspberry Pi camera.

**About**  
Real Time Emotion Detection for Low Cost Robot in ROS  
[repository](#) [issues](#) [pull requests](#) [watching](#) [forks](#)

**Files**

- main · 1 branch · 0 tags
- jmrtzma updating README
- docs adding doc Images 20 hours ago
- emotion\_detection\_ros updating models 20 days ago
- emotion\_detection\_ros\_main adding emotion\_detection\_ros last month
- LICENSE initial commit last month
- README.md updating README 20 hours ago
- requirements.txt adding requirements file last month

**README.md**

**Real Time Emotion Detection for Low Cost Robot in ROS**

**Overview**

This is a ROS package developed for emotion detection in camera images of low cost robots. The detection system has been developed in Javier Martínez's final degree project, more information [here](#).

This package has been tested under ROS Noetic in Raspberry Pi 4b under Raspberry Pi OS (buster) 32-bit. The camera that has been used is the Raspberry Pi Camera Module V2.2.

**Author:** Javier Martínez

Based on [The Extended Cohn-Kanade Dataset \(CK+\)](#)[2] this system can detect 4 emotions:

- Happy
- Surprise
- Anger
- Sadness

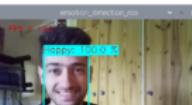
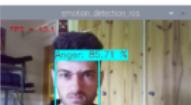
There are three models pre-trained available for use:

- KNN (K Nearest Neighbours) - Accuracy: 95% (model recommended)
- SVM (Support Vector Machine) - Accuracy: 95% (value of probability in output not available)
- MLP (Multi Layer Perceptron) - Accuracy: 95%

In [Model configuration](#) section explains how to change the model.

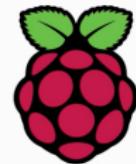
**One face examples**

[Video DEMO](#) [here](#).

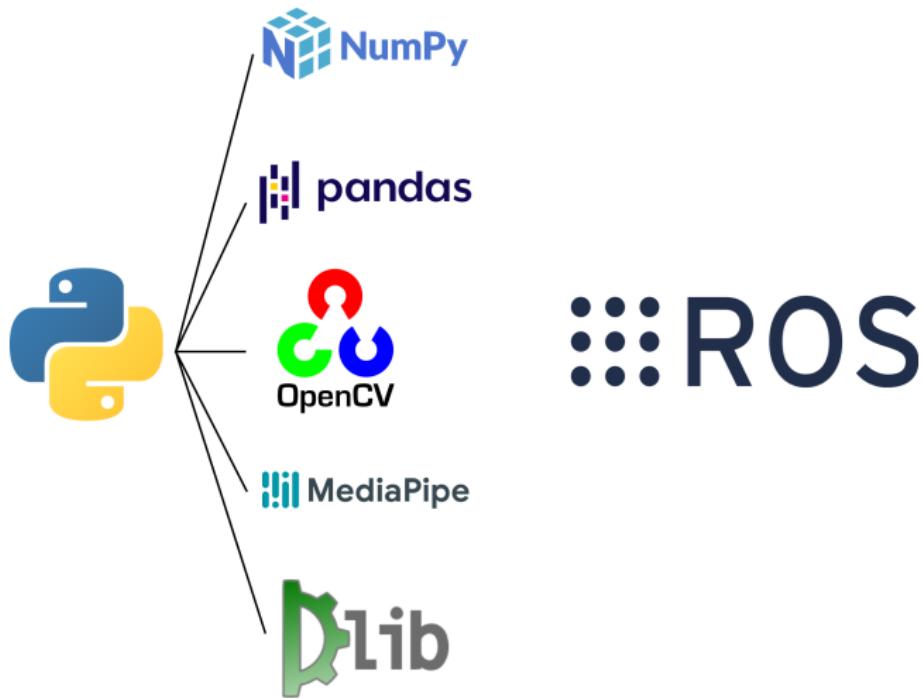
# *Plataforma de desarrollo*

# Raspberry

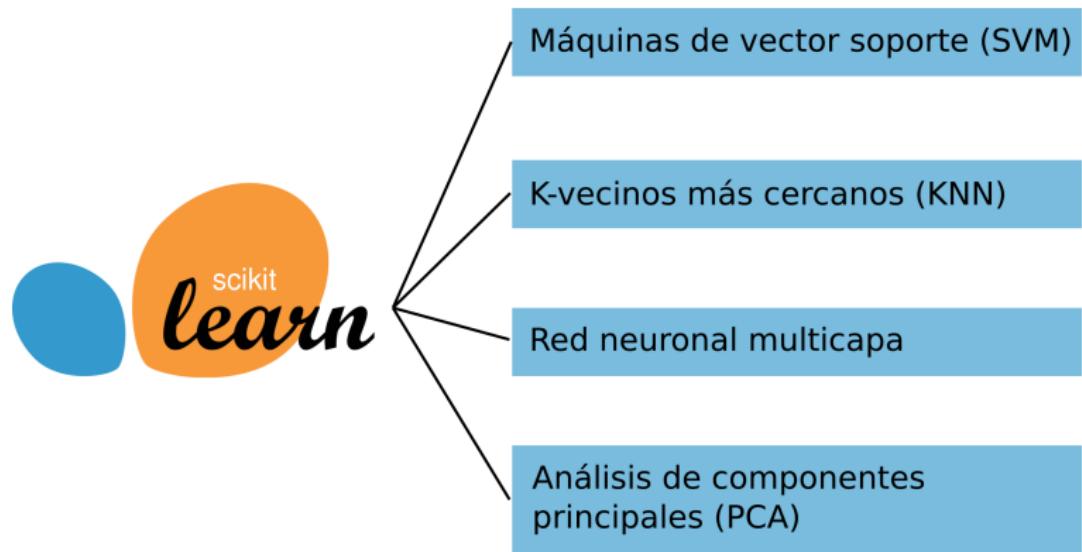


Raspberry Pi OS

# Herramientas Software

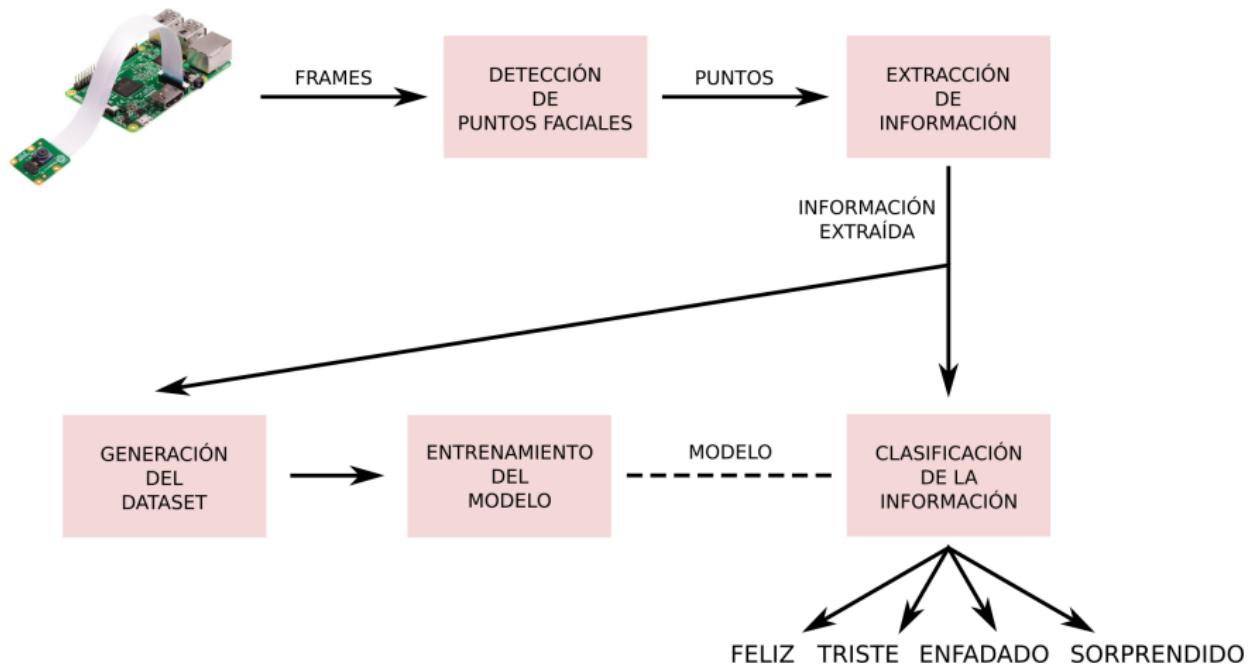


# Algoritmos de Machine Learning

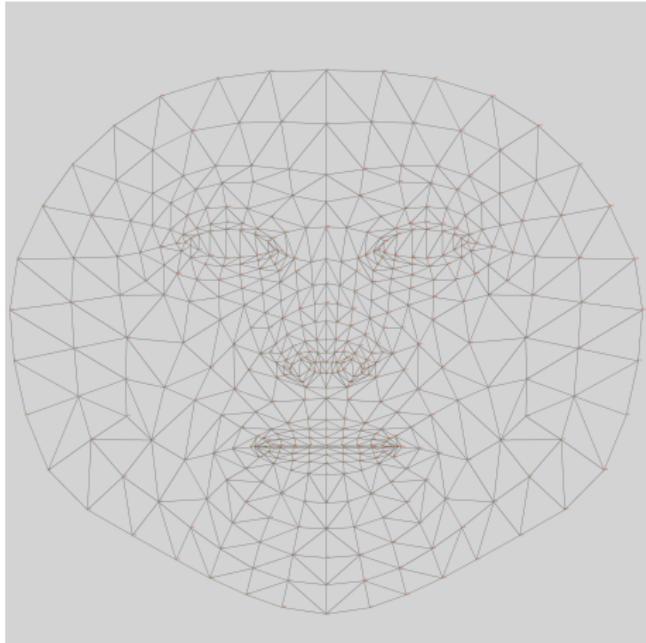


# *Sistema de detección de emociones*

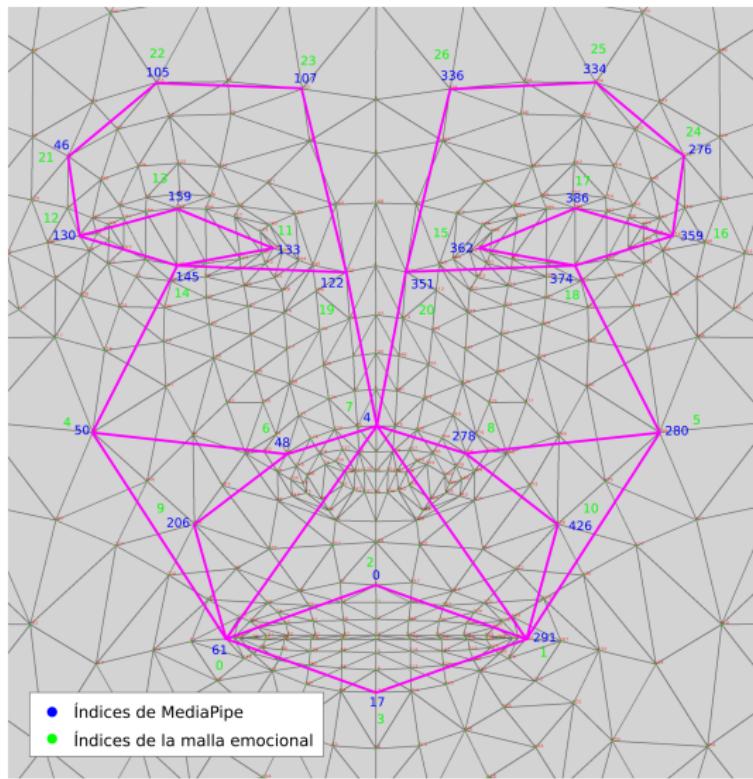
# Esquema general



# MediaPipe FaceMesh



# Malla emocional



# FACS y EMFACS

AU	Descripciones FACS
1	Interior de las cejas elevado
2	Exterior de las cejas elevado
4	Cejas bajadas
5	Párpado superior elevado
6	Mejillas elevadas
7	Párpados tensos
9	Nariz arrugada
10	Labio superior elevado
12	Comisuras de los labios elevados
15	Comisuras de los labios hacia abajo
16	Labio inferior hacia abajo
17	Barbillas elevadas
20	Labios apretados y estirados
22	Labios en forma de o
23	Labios tensos
24	Labios presionados
25	Labios separados
26	Boca abierta (mandíbula caída)
27	Boca abierta

Emoción	AU
Felicidad	6 + 12
Tristeza	1 + 4 + 15
Sorpresa	1 + 2 + 5 + 26
Miedo	1 + 2 + 4 + 5 + 7 + 20 + 26
Enfado	4 + 5 + 7 + 23
Asco	9 + 15 + 17
Desprecio	12 + 14

# The Extended Cohn-Kanade Dataset (CK+)



## Contenido

- 327 imágenes etiquetadas
- 7 clases (anger, contempt, disgust, fear, happy, sadness, surprise)

# Dataset generado

X0	X1	X2	...	X19	X20	y
54.288044	37.570711	154.655589	...	44.625203	63.887010	1.0
44.670597	35.229102	148.630240	...	47.334403	61.278073	1.0
46.613914	36.808837	161.148375	...	57.291823	64.390395	1.0
49.404349	47.407905	153.817836	...	49.880184	61.894869	1.0
42.510847	43.626048	146.891826	...	46.965439	59.971707	1.0
...	...	...	...	...	...	...
23.444336	97.667648	88.384186	...	28.011004	63.905389	7.0
24.634940	96.406625	93.413763	...	28.637606	63.551521	7.0
22.425106	105.319774	87.727242	...	30.811141	61.646881	7.0
15.966920	120.968600	60.790043	...	28.018767	56.442379	7.0
19.667632	104.568158	80.332991	...	29.088510	64.107810	7.0

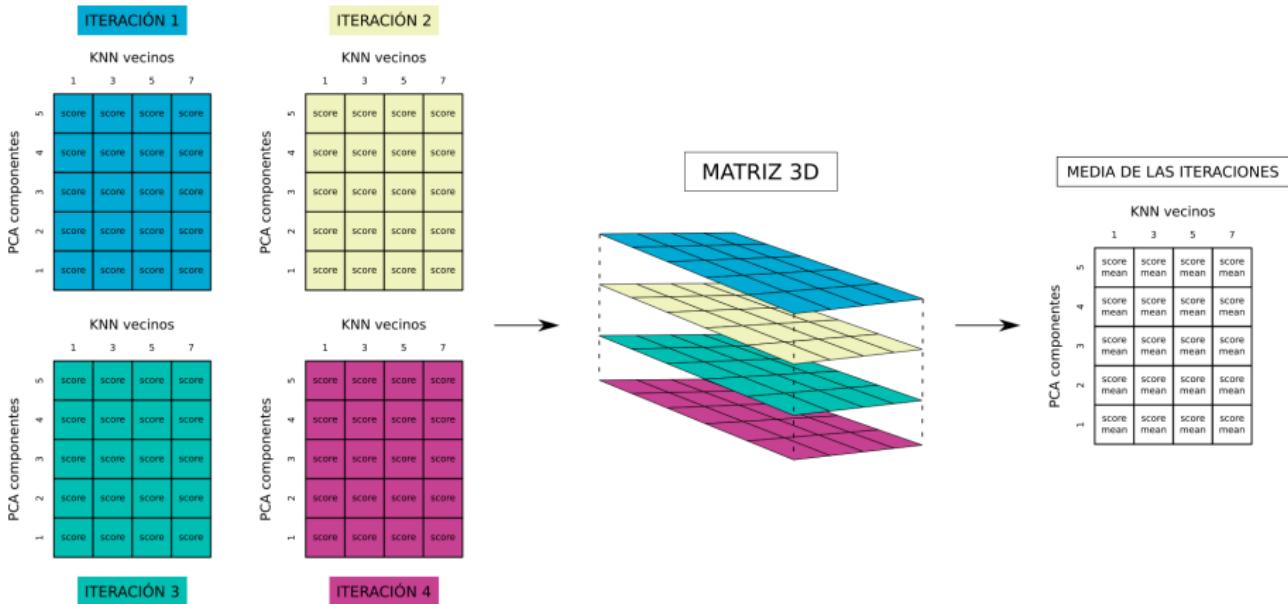
## Contenido

- 226 muestras
- 21 características
- 4 clases (anger, happy, sadness, surprise)

# Validación cruzada K-Fold Stratified de 4 pliegues

Iteración 1	Test	Train	Train	Train
Iteración 2	Train	Test	Train	Train
Iteración 3	Train	Train	Test	Train
Iteración 4	Train	Train	Train	Test

# Búsqueda de los parámetros óptimos



# Parámetros óptimos

## Intervalos de valores testeados

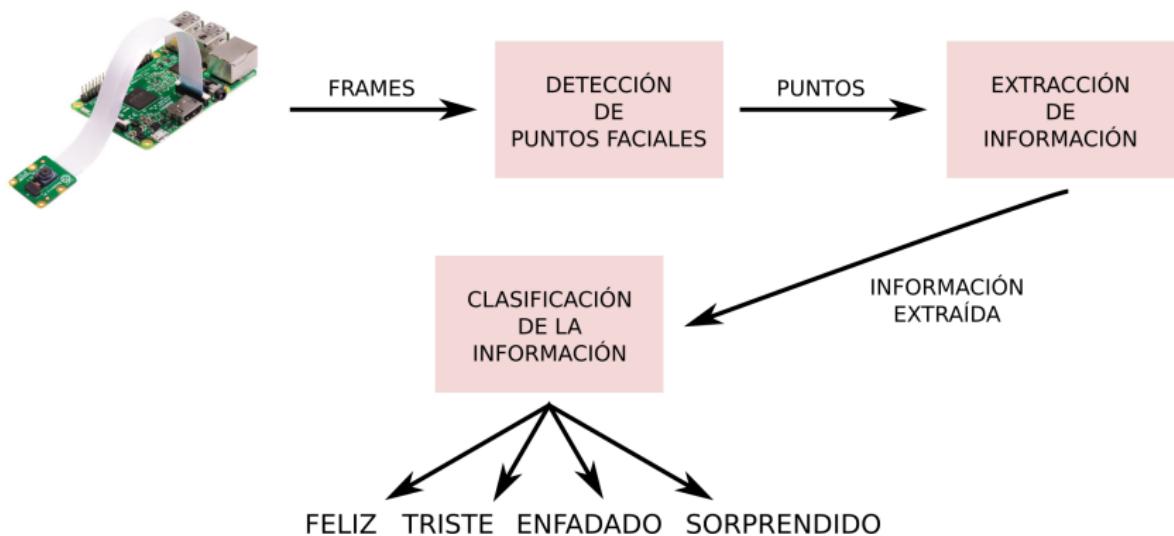
- PCA: número de *componentes* en [2, 20]
- KNN: valores impares de  $k$  en [1, 13]
- SVM: valores de  $C$  en [1, 999] con saltos de 10
- MLP: 1 capa oculta de [5, 24] *neuronas*

Clasificador y PCA	Parámetros
KNN y PCA	$k = 7$ , n_components = 11
SVM y PCA	$C = 21$ , n_components = 11
MLP y PCA	hidden_layer_sizes = (17), n_components = 11

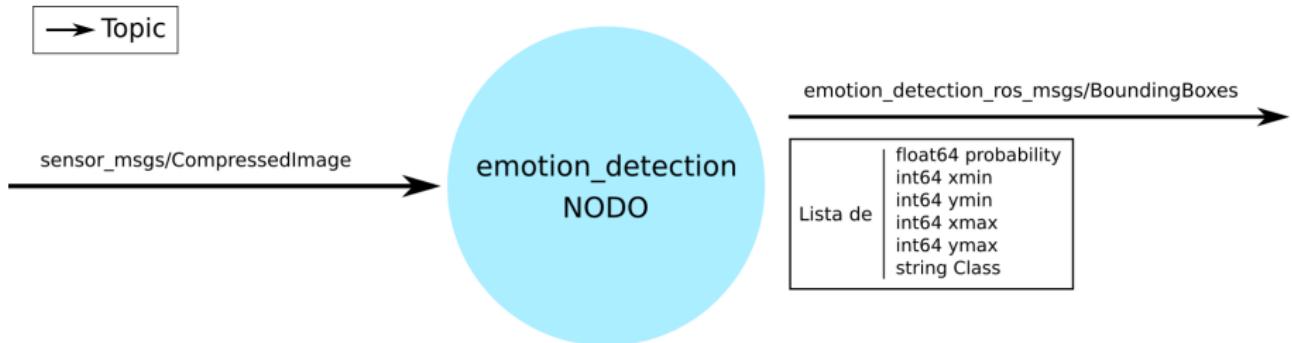
# Resultados del entrenamiento

Clasificador	Accuracy	Precision	Recall	F1-score
KNN	0.95	0.93	0.94	0.92
SVM	0.95	0.93	0.92	0.92
MLP	0.95	0.95	0.93	0.93

# Esquema de funcionamiento del sistema



# Esquema de funcionamiento del sistema en ROS

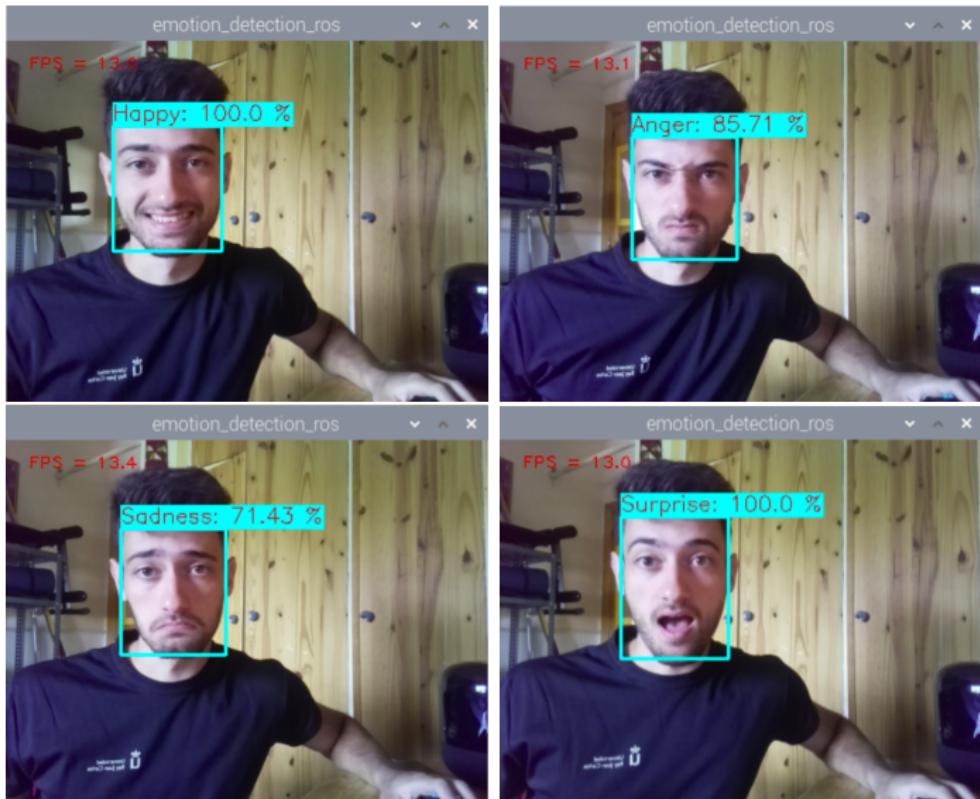


# Parámetros de configuración

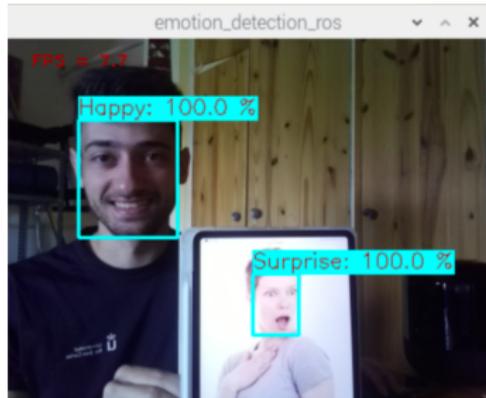
```
1 model:  
2  
3     algorithm: KNN  
4     max_num_faces: 1
```

```
1     subscribers:  
2  
3         camera_reading:  
4             topic: /raspicam_node/image/compressed  
5             queue_size: 1  
6  
7         publishers:  
8  
9             bounding_boxes:  
10                topic: /emotion_detection_ros/bounding_boxes  
11                queue_size: 1  
12                latch: False  
13  
14         image_view:  
15  
16             enable: True  
17             wait_key_delay: 1
```

# Ejemplo de funcionamiento con 1 cara



# Ejemplo de funcionamiento con 2 caras



```
---  
bounding_boxes:  
-  
    probability: 100.0  
    xmin: 198  
    ymin: 199  
    xmax: 234  
    ymax: 249  
    Class: "Surprise"  
-  
    probability: 100.0  
    xmin: 42  
    ymin: 74  
    xmax: 131  
    ymax: 177  
    Class: "Happy"  
-  
-
```

# Rendimiento

max\_num\_faces: 1

Caras detectadas	Media de FPS	Valor máximo de FPS	Valor mínimo de FPS
1	13.18	15.13	11.04

max\_num\_faces: 2

Caras detectadas	Media de FPS	Valor máximo de FPS	Valor mínimo de FPS
1	11.11	11.97	9.18
2	7.53	8.35	6.96

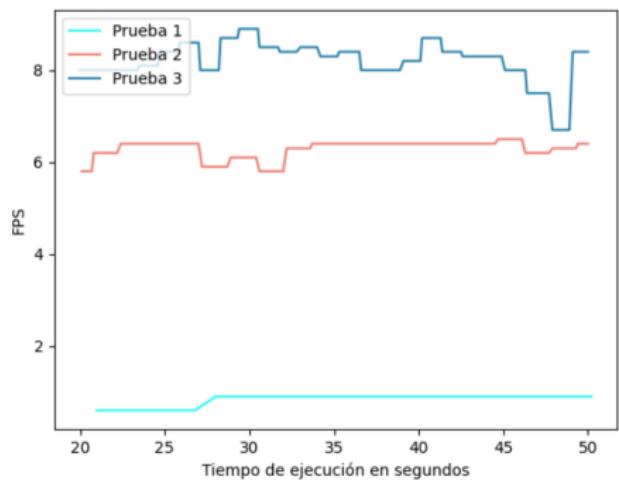
max\_num\_faces: 3

Caras detectadas	Media de FPS	Valor máximo de FPS	Valor mínimo de FPS
1	11.05	11.72	9.41
2	6.77	7.61	6.15
3	5.37	6.93	5.02

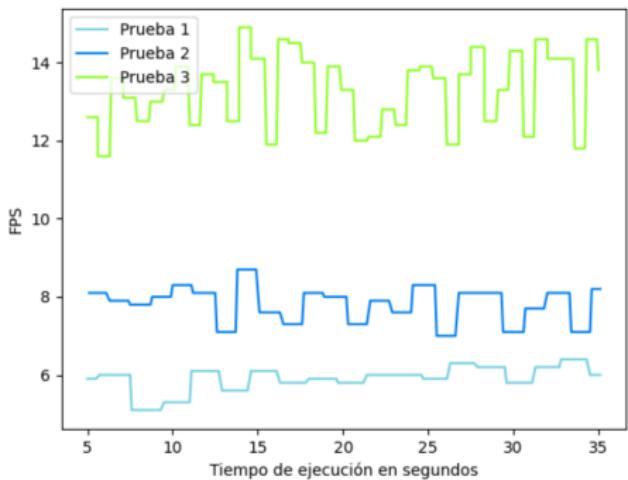
# *Estudios de optimización*

# Estudio de rendimiento para dlib y MediaPipe

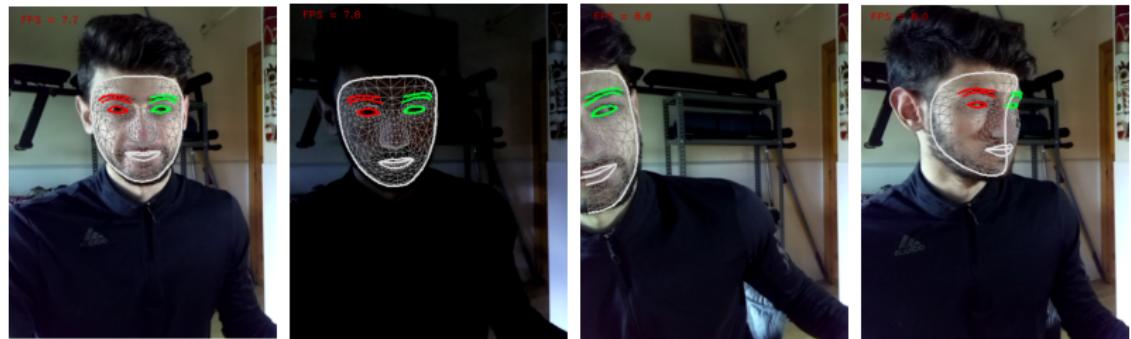
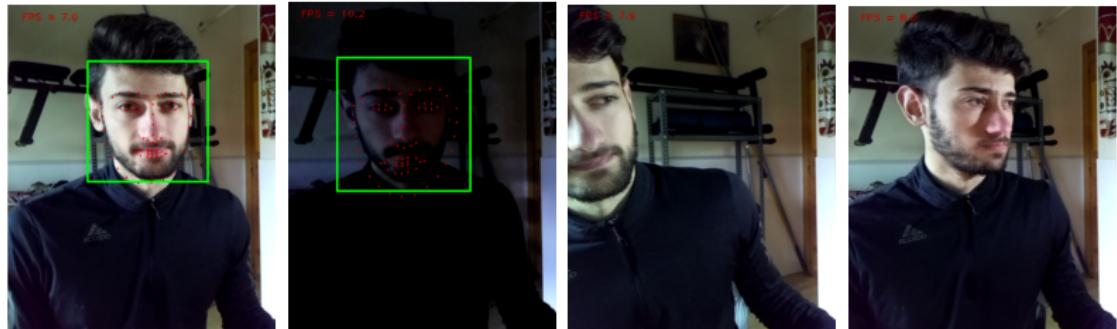
## dlib



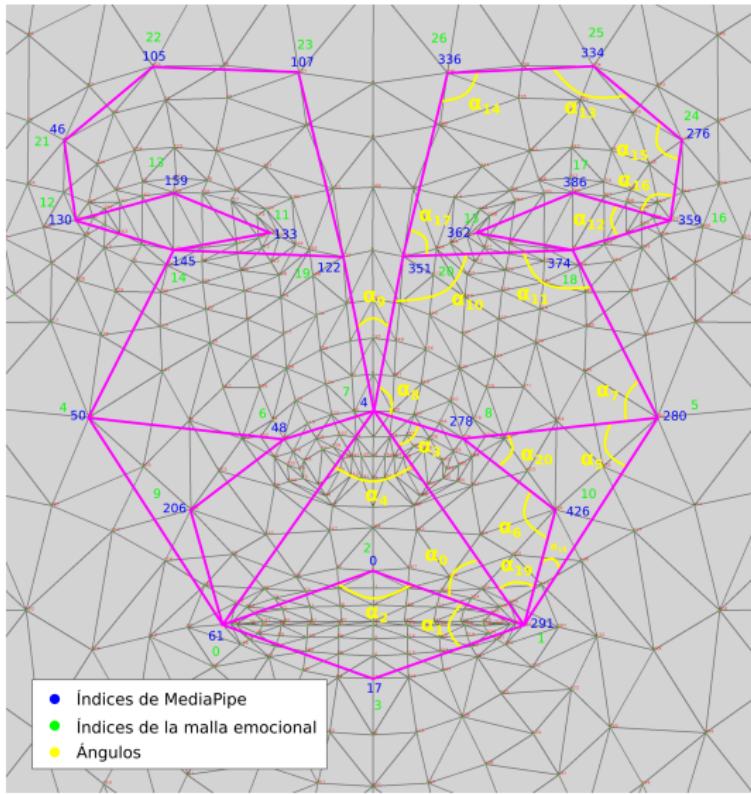
## MediaPipe



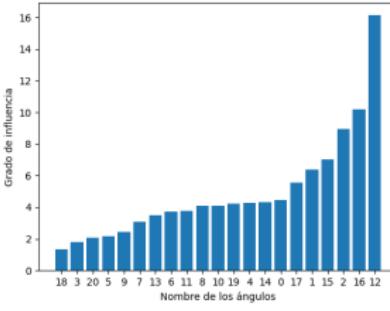
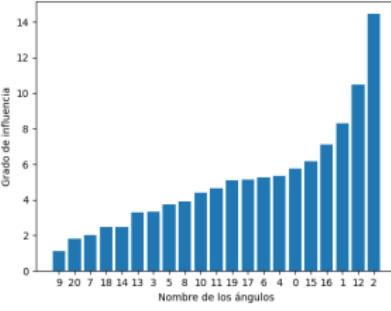
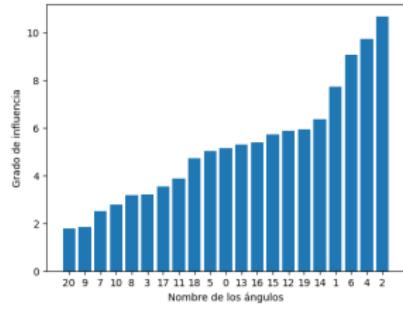
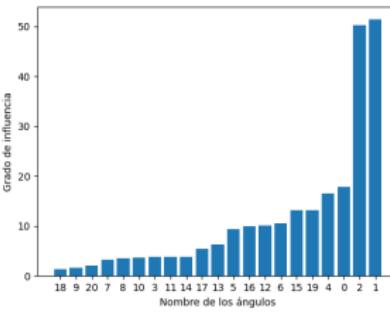
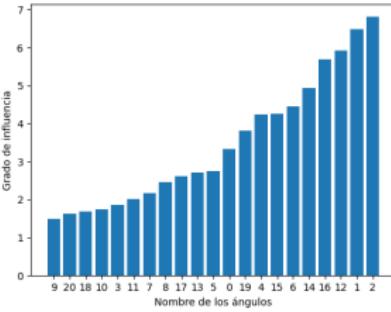
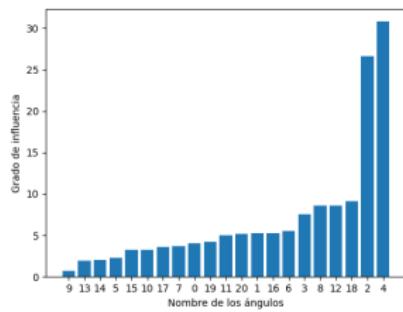
# Estudio de precisión para dlib y MediaPipe



# Estudio de ángulos más influyentes en cada emoción



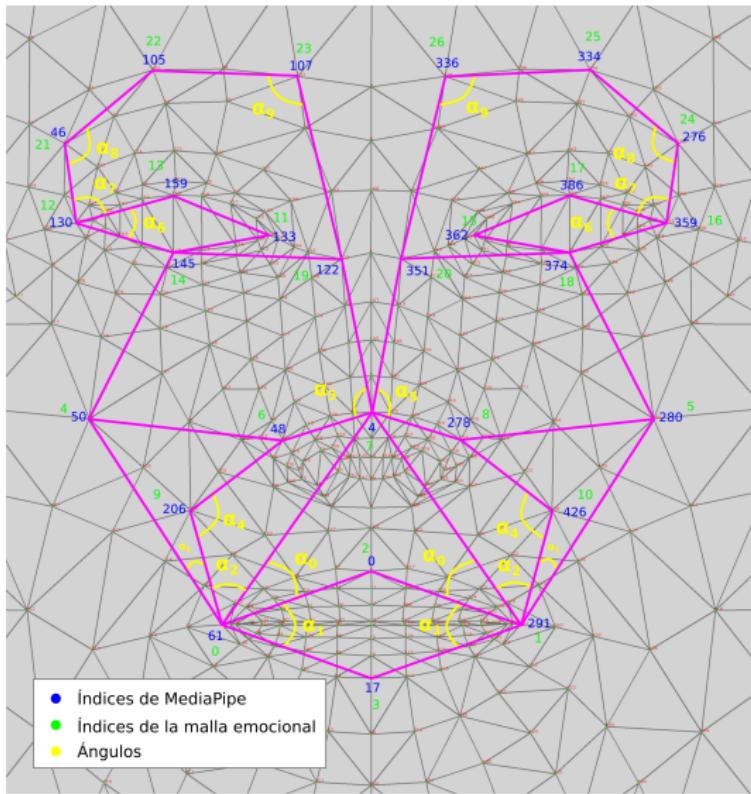
# Resultados del estudio de ángulos más influyentes



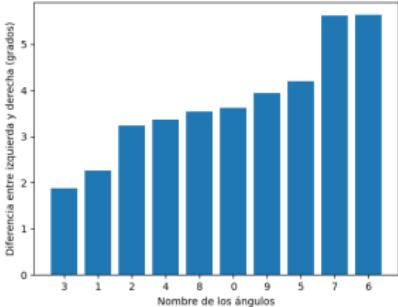
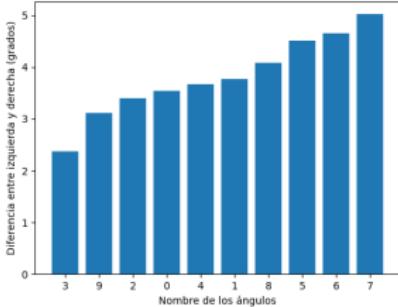
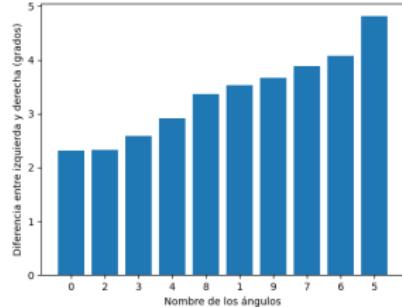
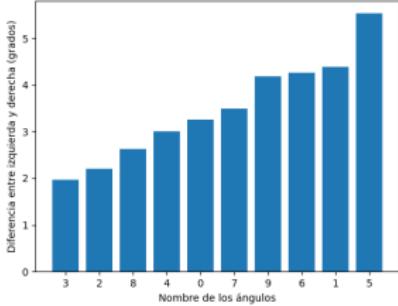
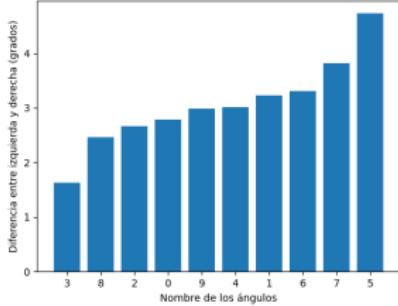
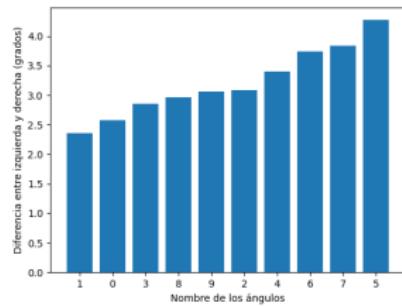
# Resultados del estudio de ángulos más influyentes

Emoción	Ángulos influyentes
Felicidad	4, 2, 18, 12, 8
Tristeza	2, 1, 12, 16, 14
Sorpresa	1, 2, 0, 4, 19
Miedo	2, 4, 6, 1, 14
Enfado	2, 12, 1, 16, 15
Asco	12, 16, 2, 15, 1
Desprecio	2, 1, 4, 0, 19

# Estudio de simetría de las emociones



# Resultados del estudio de simetría



# Búsqueda del dataset con más rendimiento

## Resultados entrenamiento dataset1

Clasificador	Accuracy	Precision	Recall	F1-score
KNN	0.82	0.76	0.74	0.74
SVM	0.84	0.79	0.78	0.77
MLP	0.82	0.78	0.78	0.77

## Resultados entrenamiento dataset2

Clasificador	Accuracy	Precision	Recall	F1-score
KNN	0.83	0.79	0.75	0.75
SVM	0.84	0.80	0.77	0.78
MLP	0.84	0.81	0.80	0.80

# Resultados del entrenamiento con SVM de cada iteración de K-Fold usando el dataset2

Clase	Precision	Recall	F1-score
Enfado	0.67	0.50	0.57
Desprecio	0.67	0.50	0.57
Asco	0.76	0.87	0.81
Miedo	0.67	1.00	0.80
Felicidad	0.94	0.94	0.94
Tristeza	0.83	0.71	0.77
Sorpresa	0.95	0.95	0.95

Clase	Precision	Recall	F1-score
Enfado	0.71	0.45	0.56
Desprecio	0.57	0.80	0.67
Asco	0.76	0.93	0.84
Miedo	1.00	0.71	0.83
Felicidad	0.85	1.00	0.92
Tristeza	0.83	0.71	0.77
Sorpresa	1.00	0.95	0.98

Clase	Precision	Recall	F1-score
Enfado	0.75	0.82	0.78
Desprecio	0.67	0.80	0.73
Asco	0.82	0.93	0.87
Miedo	0.80	0.67	0.73
Felicidad	0.94	0.94	0.94
Tristeza	1.00	0.57	0.73
Sorpresa	0.95	0.95	0.95

Clase	Precision	Recall	F1-score
Enfado	0.67	0.55	0.60
Desprecio	0.50	0.25	0.33
Asco	0.67	0.80	0.73
Miedo	1.00	0.67	0.80
Felicidad	1.00	1.00	1.00
Tristeza	0.50	0.71	0.59
Sorpresa	1.00	1.00	1.00

# *Conclusiones*

## Objetivo principal cumplido

Desarrollar una herramienta de reconocimiento de emociones que sea capaz de funcionar en un sistema robótico de bajo coste bajo el entorno ROS.

## Subobjetivos cumplidos

- Estudiar el estado del arte y elegir la técnica más óptima.
- Optimizar y adaptar la técnica escogida en nuestra plataforma.
- Generar un dataset de valor.
- Realizar el entrenamiento del modelo.
- Integrar el sistema en ROS.

## Líneas futuras

- Implementar aplicaciones de HRI que se ayuden de esta herramienta.
- Portar el sistema a otras plataformas.
- Añadir nuevas funcionalidades a la herramienta.

# Sistema de detección de emociones faciales mediante técnicas de Machine Learning adaptado a ROS para un robot de bajo coste basado en Raspberry Pi

Javier Martínez Madruga

[j.martinezma.2018@alumnos.urjc.es](mailto:j.martinezma.2018@alumnos.urjc.es)



Trabajo fin de grado

28 de Junio de 2022