

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Bachillerato en Ingeniería en Computación



IC-8046 Introducción al Reconocimiento de Patrones

Profesor: Ing. Eduardo A. Canessa Montero, M.Sc.

Proyecto Final: Primer Etapa

Integrantes:

Ericka Céspedes Moya 2017239557

Esteban Alonso González Matamoros 2018113111

Fecha de Entrega: 22 de octubre

II Semestre, 2021

# Índice

<b>Introducción</b>	<b>3</b>
<b>Diseño Propuesto</b>	<b>4</b>
Estructuración del diseño basado en Diseño Modular	4
Diagrama de bloques	4
Entradas/Sensores	4
Extractor de características	4
Metodologías a utilizar	4
Clasificador en Cascada	4
Archivos de entrenamiento	5
Reconocimiento facial Face ++ para Python	5
Desarrollo en espiral	6
Diagramas de flujo	7
Etapa 2	7
Etapa 3	8
<b>Resultados esperados</b>	<b>9</b>
<b>Recomendaciones</b>	<b>9</b>
<b>Literatura consultada</b>	<b>10</b>

## **Introducción**

En este informe se detalla la propuesta para el proyecto de Introducción al Reconocimiento de Patrones. La propuesta consiste en una aplicación para identificar rostros humanos y reconocer las emociones por medio de expresiones faciales. Se incluye el diseño propuesto (la estructuración del diseño basado en Diseño Modular y la descripción de cada módulo), un resumen de los resultados esperados y las recomendaciones para las siguientes etapas.

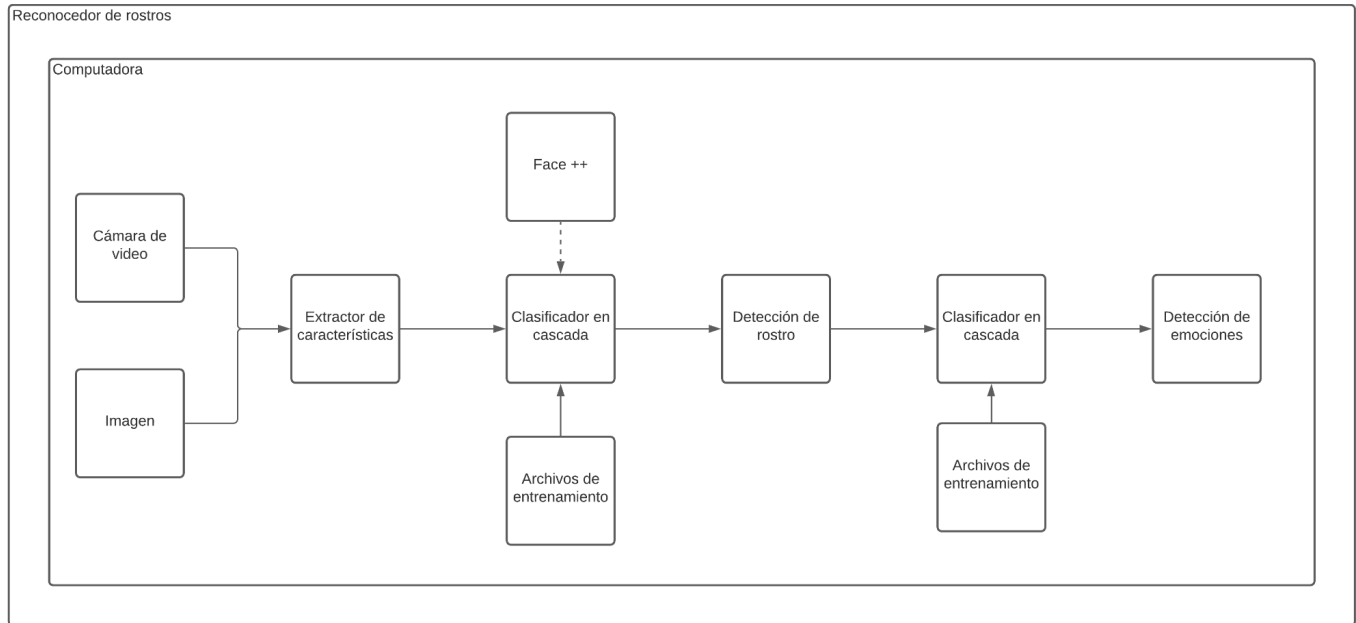
En la primera etapa se va a diseñar la solución para el problema planteado utilizando la Metodología de Diseño Modular y Desarrollo en Espiral. Además, se debe investigar las herramientas y al menos dos metodologías que se están utilizando en la actualidad para reconocimiento de rostros humanos y el reconocimiento de emociones mediante expresiones faciales. Para identificar rostros se utilizará Python y OpenCV para implementar el sistema y como metodología se utilizará Clasificación en Cascada y Deep Learning (archivos de entrenamiento). Además, se verá la utilización de Face++ que se usa para reconocer rostros en Python, pero funciona mejor en línea.

. La segunda etapa consta de desarrollar un sistema de reconocimiento de rostros humanos en imágenes y video, de forma tal que el sistema pueda contar cuantos rostros humanos aparecen delante de una cámara de forma constante, desde el momento en que aparezcan. La tercera etapa se basa en el desarrollo de la segunda etapa. Se planea diseñar un reconocedor de emociones humanas basado en el reconocimiento de rostros humanos y expresiones faciales. Una vez reconocidos los rostros, se utilizará nuevamente Clasificación en Cascada y Deep Learning (archivos de entrenamiento) para reconocer emociones de expresiones faciales.

## Diseño Propuesto

### Estructuración del diseño basado en Diseño Modular

#### Diagrama de bloques



#### **Entradas/Sensores**

Cámara de video de la computadora o la imagen a analizar

#### **Extractor de características**

Cámara de video de la computadora o la imagen a analizar

#### **Metodologías a utilizar**

##### **Clasificador en Cascada**

La clase encapsula el mecanismo de detección de destino, es decir, el mecanismo de ventana deslizante más clasificador en cascada. La estructura de datos incluye dos partes principales: Data y FeatureEvaluator. Los datos almacenan los datos del clasificador cargados del archivo xml obtenido mediante capacitación; y FeatureEvaluator trata sobre la carga, el almacenamiento y el cálculo de las características.

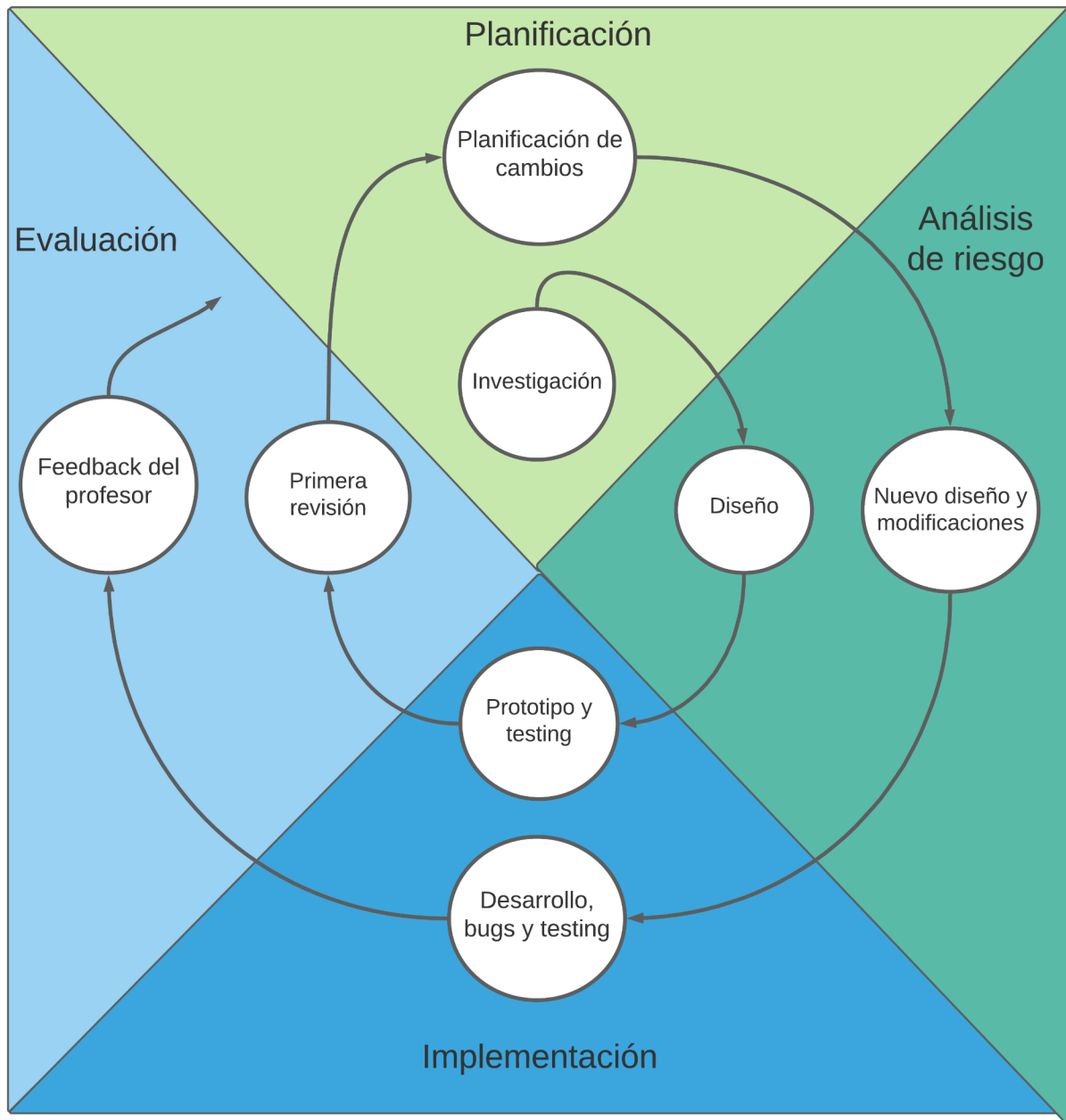
## **Archivos de entrenamiento**

El archivo de entrenamiento utilizado aquí es `haarcascade_frontalface_default.xml` provisto por defecto en OpenCV. En cuanto a los principios específicos de Haar y LBP, puede entenderse como los datos de características de la cara. Para las emociones, también se utilizarían imágenes de entrenamiento con las emociones clasificadas.

## **Reconocimiento facial Face ++ para Python**

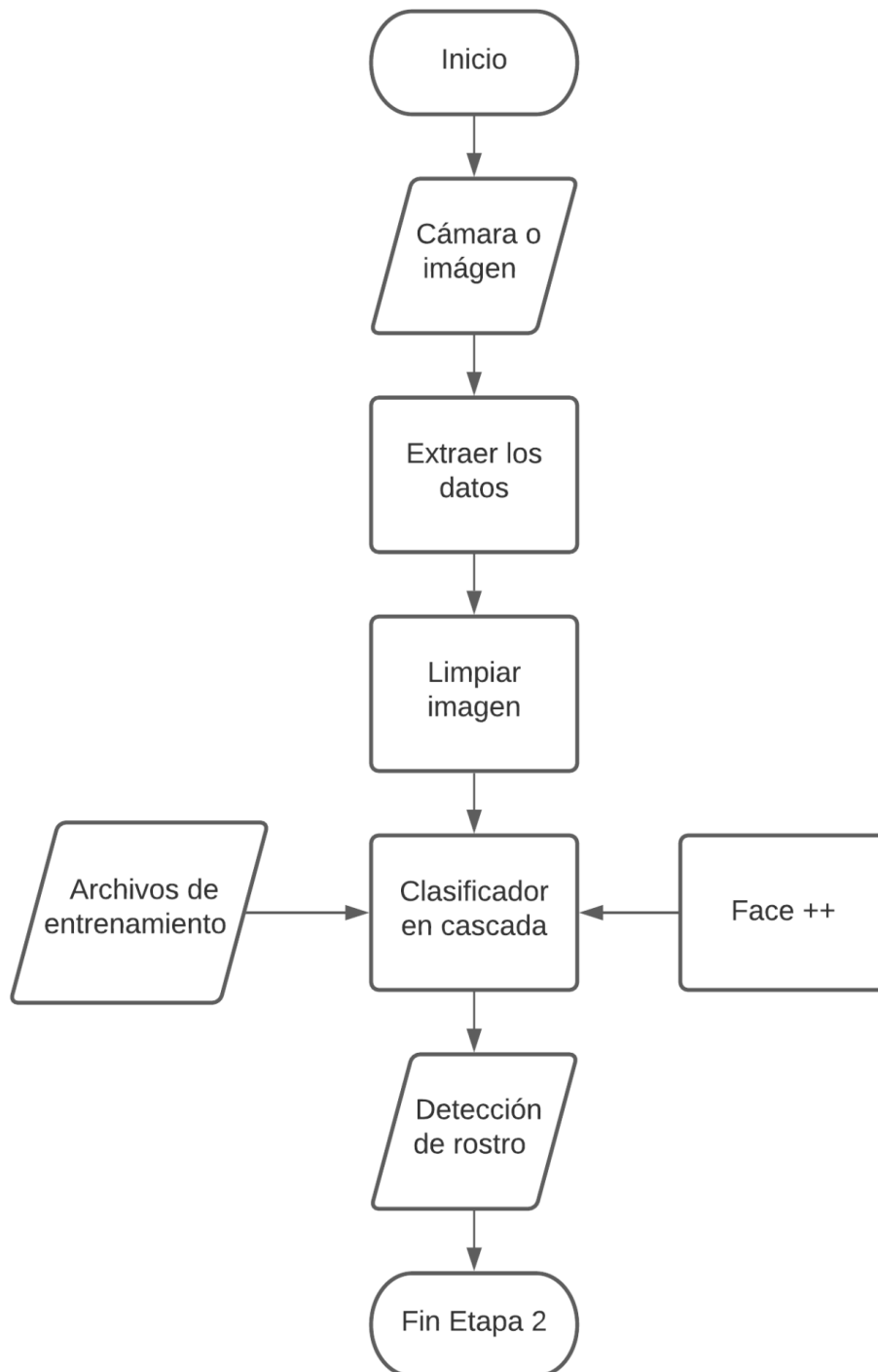
Face ++ se especializa en reconocimiento facial. La información devuelta es completa y fácil de llamar. Sin embargo, solo está disponible en línea, y el SDK sin conexión no puede proporcionar algunas facilidades en tiempo real. Por lo tanto, se agregará al clasificador como extra para que funcione mejor cuando esté en línea, pero no dependerá solo de eso.

## Desarrollo en espiral

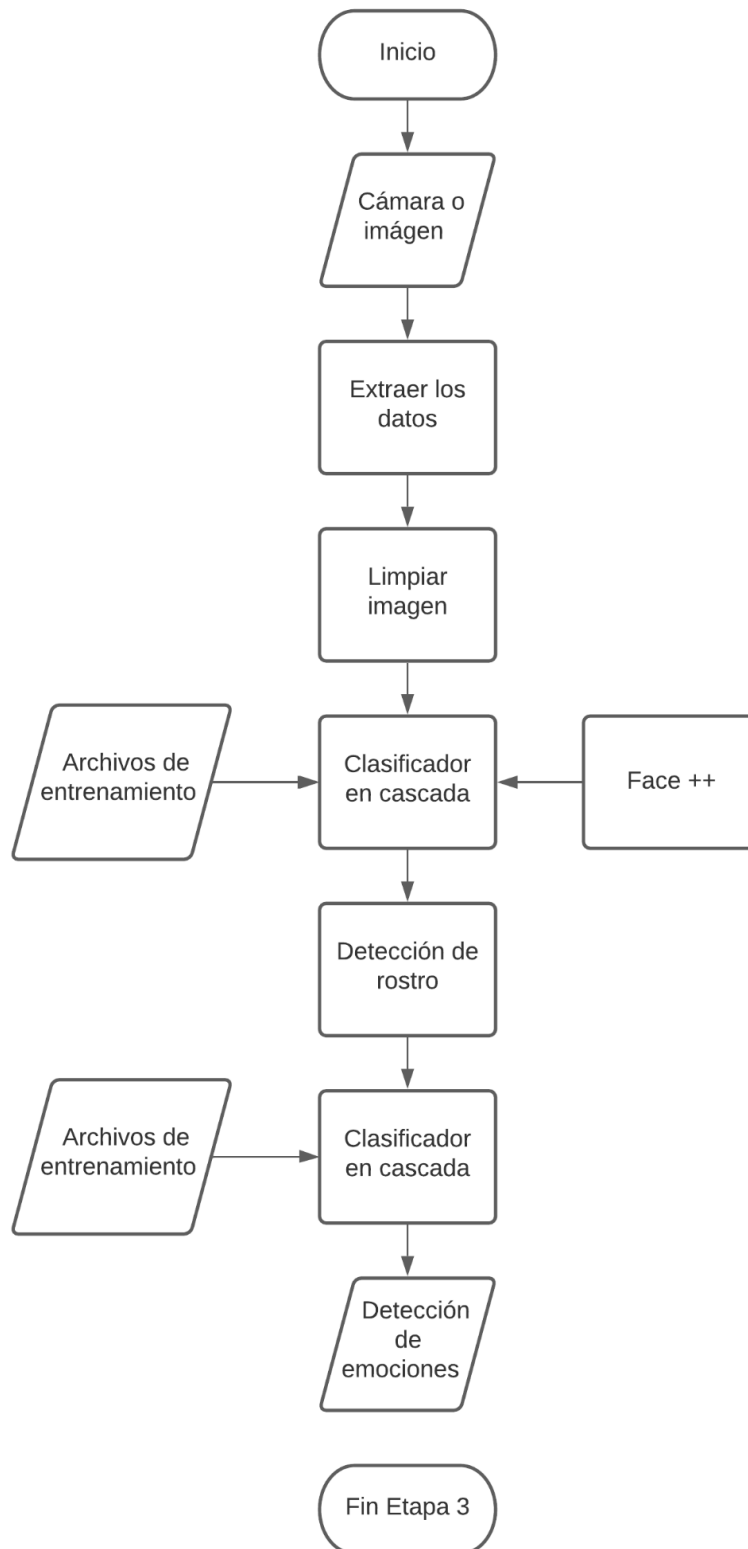


## Diagramas de flujo

### **Etapas 2**



### Etapa 3





## **Resultados esperados**

- En la segunda etapa se planea tener un sistema de reconocimiento de rostros humanos en imágenes y video, de forma tal que el sistema pueda contar cuantos rostros humanos aparecen delante de una cámara de forma constante, desde el momento en que aparezcan, para ello.
- En la tercera etapa, basándose en el desarrollo de la segunda etapa de este proyecto, se desea contar con un sistema que reconoce emociones humanas basado en el reconocimiento de rostros humanos y expresiones faciales.

## **Recomendaciones**

- Utilizar Python para programar el proyecto, ya que es un lenguaje sumamente sencillo de comprender y utilizar, además, su comunidad se mantiene sumamente activa permitiendo que sea más sencillo encontrar soluciones a problemas debido a que algún otro usuario ya los ha experimentado, también permite obtener ayuda más fácilmente por medio de foros, redes sociales o conocidos. También posee el beneficio de que en este lenguaje se suelen programar redes neuronales e Inteligencias artificiales que pueden servir para detección de patrones en imágenes y videos.
- Utilizar la biblioteca de OpenCV en Python que ofrece muchas funcionalidades para manipulación de imágenes.
- Utilizar las metodologías de clasificadores en cascada y archivos de entrenamiento para reconocer rostros, así como emociones.
- Utilizar Face ++ para tener un reconocedor más completo cuando se tiene conexión a Internet.
- Utilizar los recursos que se tiene a mano, en este caso, una computadora y su cámara de video.

## **Literatura consultada**

- A. Rosebrock, "OpenCV Haar Cascades," *PyImageSearch*, 17-Apr-2021. [Online]. Recuperado de: <https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/>.
- F. Tian, "Detección de rostro," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/83991134833/>.
- F. Tian, "Detección de rostros con código Python," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/9624563954/>.
- F. Tian, "Detección de rostros de la cámara Python OpenCV3," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/43641484426/>.
- F. Tian, "Detección de rostros Python + OpenCV," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/1972911136/>.
- F. Tian, "Detección de rostros y reconocimiento de rostros basados en python y face ++," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/4425247087/>.
- F. Tian, "Python realización súper simple del reconocimiento de emociones faciales humanas," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/87201816165/>. [Accessed: 22-Oct-2021].
- F. Tian, "Reconocimiento de emociones OpenCV (implementado en Python, súper simple)," *programador clic*. [Online]. Recuperado de: <https://programmerclick.com/article/43501075874/>.
- G. S. Behera, "Face detection with Haar Cascade," *Medium*, 29-Dec-2020. [Online]. Recuperado de: <https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08>.

OpenCV, "opencv/data/haarcascades/." *GitHub*, 12-Apr-2020 [Online]. Recuperado de:  
<https://github.com/opencv/opencv/tree/master/data/haarcascades>.