



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“EVALUACIÓN E IMPLEMENTACIÓN DE TÉCNICAS DE CLASIFICACIÓN  
DE PATRONES ORIENTADOS A DISPOSITIVOS DE BAJO PODER  
COMPUTACIONAL”

**INFORME DE MATERIA DE GRADUACIÓN**

Previa a la obtención del Título de:  
**INGENIERÍA EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS  
MULTIMEDIA**

**Presentado por:**  
Yuri Elizabeth Cosquillo Macías

**GUAYAQUIL - ECUADOR**

**AÑO: 2015**

## **AGRADECIMIENTOS**

Agradezco a Dios por su infinito amor, por mostrarme cada día que siempre hay una oportunidad para empezar a pesar de las dificultades que se presenten. Quisiera agradecerles a mis padres. Por sus consejos, por su incondicional apoyo, por ser una gran luz para mi vida, especialmente a mi madre por su compañía, amor y paciencia. Agradezco a mis profesores que impartieron sus enseñanzas a lo largo de mi vida politécnica. Y finalmente a mi tutor de tesis Dr. Dennis Romero López por su tiempo y dedicación.

## DEDICATORIA

Dedico este trabajo primeramente a Dios quién supo guiarme hacia este camino, a mi madre por su incansable lucha y esfuerzo que me permite culminar una meta más y el comienzo de muchas, a mis compañeros de estudio que se mantuvieron cerca brindándome su apoyo incondicional.

## TRIBUNAL DE SUSTENTACIÓN

---

PhD. Dennis Romero López  
PROFESOR DE LA MATERIA DE GRADUACIÓN

---

PhD. Carmen Vaca Ruiz  
PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

## DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este informe me corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL).

---

Yuri Elizabeth Cosquillo Macías

## RESUMEN

Dentro de la temática del proyecto titulado “Evaluación e implementación de técnicas de clasificación de patrones orientados a dispositivos de bajo poder computacional”, el presente trabajo describe la evaluación de técnicas de clasificación y la implementación de una de ellas aplicándola al desarrollo de un sistema colaborativo para la identificación de especímenes Lepidópteros (mariposas), reconocidas por reservas ecológicas Ecuatorianas. Este proyecto busca contribuir con estudios en el área de biología acerca de géneros de mariposas conocidas y a su vez facilitar el registro de especímenes no reconocidos, por medio de la ayuda de expertos a nivel mundial.

Este proyecto es de naturaleza multidisciplinario y cuenta actualmente con el apoyo del *Parque Nacional Sangay*, el cual facilitó el banco de datos con el cual tuvo lugar este estudio. Mediante la extracción de características de imágenes basadas en forma, ocho géneros de mariposas son identificados automáticamente mediante el uso de algoritmos de clasificación. Con el proceso de entrenamiento del algoritmo, el computador (dispositivo embebido) adquiere la capacidad de identificar determinados géneros de mariposas, lo cual es usado en este trabajo para realizar una evaluación de conocimientos a estudiantes de biología, quienes serán cuestionados acerca del género a la cual pertenece una mariposa, por medio de una aplicación para dispositivo móvil.

Cuando el estudiante termina la evaluación, el género escogido es posteriormente visualizado por un profesional del área. Este último procede a corroborar si la respuesta dada por el estudiante acerca de la mariposa presentada fue correcta o incorrecta. La confirmación dada por el experto lanza un evento en la aplicación del estudiante con el fin de retroalimentarlo y también, para reforzar el conocimiento que el computador tiene acerca del género que fue escogido.

# ÍNDICE GENERAL

|   |          |
|---|----------|
| RESUMEN   | vi       |
| INDICE GENERAL                                    | vii      |
| INDICE DE TABLAS                                  | x        |
| INDICE DE FIGURAS                                 | xi       |
| INTRODUCCIÓN                                      | xiii     |
| <b>1 DESCRIPCIÓN Y JUSTIFICACIÓN DEL PROYECTO</b> | <b>1</b> |
| 1.1 Descripción del problema . . . . .            | 1        |
| 1.2 Justificación del proyecto . . . . .          | 2        |
| 1.3 Solución propuesta . . . . .                  | 4        |
| 1.4 Objetivos . . . . .                           | 5        |
| 1.4.1 Objetivo General . . . . .                  | 5        |
| 1.4.2 Objetivos Específicos . . . . .             | 5        |
| 1.5 Organización del documento . . . . .          | 6        |
| <b>2 REVISIÓN BIBLIOGRÁFICA</b>                   | <b>8</b> |
| 2.1 Pre procesamiento de imágenes . . . . .       | 8        |
| 2.2 Extracción de Características . . . . .       | 9        |
| 2.3 Algoritmos de Clasificación . . . . .         | 12       |

|          |  |           |
|----------|--|-----------|
| 2.3.1    | K-vecinos más cercanos (KNN) . . . . .                               | 15        |
| 2.3.2    | Perceptrón Multicapa (MLP) . . . . .                                 | 16        |
| 2.3.3    | Máquina de Soporte Vectorial (SVM) . . . . .                         | 17        |
| <b>3</b> | <b>INSTALACIÓN Y CONFIGURACIÓN</b>                                   | <b>18</b> |
| 3.1      | Introducción . . . . .   | 18        |
| 3.2      | Hardware/Software . . . . .  | 19        |
| 3.2.1    | Instalación del Hardware . . . . .                                   | 22        |
| 3.2.2    | Instalación de Software . . . . .                                    | 23        |
| 3.2.3    | Arquitectura básica del sistema . . . . .                            | 24        |
| <b>4</b> | <b>DESARROLLO Y IMPLEMENTACIÓN</b>                                   | <b>26</b> |
| 4.1      | Introducción . . . . .   | 26        |
| 4.2      | Desarrollo del Sistema . . . . .                                     | 26        |
| 4.2.1    | Conversión a escala de grises . . . . .                              | 27        |
| 4.2.2    | Binarización . . . . .   | 27        |
| 4.2.3    | Extracción de características . . . . .                              | 27        |
| 4.2.4    | Algoritmo de clasificación de mariposas . . . . .                    | 28        |
| 4.3      | Desarrollo del Aplicativo Móvil . . . . .                            | 28        |
| 4.3.1    | Introducción . . . . .   | 28        |
| 4.3.2    | Análisis de Requerimientos y Diseño de la plataforma móvil . . . . . | 30        |
| 4.4      | Colaboración Alumno / Experto . . . . .                              | 31        |
| 4.5      | Implementación del Software . . . . .                                | 34        |
| <b>5</b> | <b>ANÁLISIS DE RESULTADOS</b>  | <b>36</b> |
| 5.1      | Resultados . . . . .   | 36        |
|          | <b>CONCLUSIONES Y RECOMENDACIONES</b>                                | <b>44</b> |
|          | <b>BIBLIOGRAFÍA</b>  | <b>46</b> |



ANEXOS

# ÍNDICE DE TABLAS

|           |   |    |
|-----------|---|----|
| Tabla 4.1 | Pantallas del aplicativo móvil. . . . . | 31 |
| Tabla 5.1 | Caso USR-001 . . . . .                  | 49 |
| Tabla 5.2 | Caso USR-002 . . . . .                  | 50 |
| Tabla 5.3 | Caso USR-003 . . . . .                  | 51 |
| Tabla 5.4 | Caso USR-004 . . . . .                  | 52 |
| Tabla 5.5 | Caso USR-005 . . . . .                  | 53 |

# ÍNDICE DE FIGURAS

|          |   |    |
|----------|---|----|
| Fig. 1.1 | Estructura del sistema de evaluación de conocimientos sobre especímenes de mariposas. . . . . | 5  |
| Fig. 2.1 | Similitud entre neurona biológica y artificial. . . . .                                       | 13 |
| Fig. 3.1 | BeagleBone Black. . . . .   | 20 |
| Fig. 3.2 | Dispositivo móvil Samsung SIII. . . . .   | 20 |
| Fig. 3.3 | Logo de OpenCV. . . . .   | 21 |
| Fig. 3.4 | Arquitectura básica del sistema . . . . .   | 25 |
| Fig. 4.1 | Desarrollo en Android Studio. . . . .   | 29 |
| Fig. 4.2 | Aplicativo en Android Studio. . . . .   | 30 |
| Fig. 4.3 | Pantalla de ingreso al sistema web. . . . .   | 32 |
| Fig. 4.4 | Pantalla del listado de especímenes categorizados. . . . .                                    | 33 |
| Fig. 4.5 | Pantalla de inicio al sistema web. . . . .  | 33 |
| Fig. 4.6 | Pantalla del listado de evaluaciones. . . . .   | 34 |
| Fig. 4.7 | Desarrollo de la Implementación del sistema. . . . .  | 34 |
| Fig. 5.1 | Imagen original. . . . .  | 37 |
| Fig. 5.2 | Imagen en escala de grises. . . . .   | 37 |
| Fig. 5.3 | Imagen binarizada. . . . .  | 38 |
| Fig. 5.4 | Visualización de patrones de mariposas. . . . .   | 39 |
| Fig. 5.5 | Visualización de patrones de mariposas, utilizando kernel T-Student. . . . .                  | 40 |

|          |  |    |
|----------|--|----|
| Fig. 5.6 | Resultados de los algoritmos de clasificación. . . . .     | 41 |
| Fig. 5.7 | Tiempos de extracción y clasificación de patrones. . . . . | 42 |

## INTRODUCCIÓN

En la actualidad los dispositivos móviles constituyen herramientas de uso diario para personas de todas partes del mundo, esto ha venido acompañado de los avances de tecnologías multimedia y han permitido no solo facilitar la comunicación entre personas sino también la colaboración en entornos educativos, laborales, familiares, entre otros.

En el entorno educativo, son muy diversos los recursos con los que actualmente los profesores cuentan para alcanzar los objetivos de aprendizaje deseados, encontrándose las aplicaciones para dispositivos móviles entre las más adecuadas para lograr situaciones colaborativas. En biología, por ejemplo, los estudiantes aprenden a identificar especies como parte de su formación profesional y, para esto, es necesario contar con herramientas que faciliten el resumen de información que será presentada a los estudiantes, lo cual representa un reto para los profesores quienes deben resumir grandes cantidades de información sobre clases, familias, géneros y los detalles que diferencian a cada uno de ellas.

Diferentes herramientas han sido desarrolladas para contribuir con la clasificación de especies de animales y plantas de forma automática, impulsados por técnicas de reconocimiento de patrones basados en características visuales, obtenidos a partir de imágenes. Sin embargo, dada la complejidad del tema, es indispensable la incorporación de conocimiento experto que permita reforzar la identificación automática de biodiversidad.

Por tal razón, este proyecto de materia de graduación abordó el desarrollo de un sistema colaborativo para la identificación de especies de mariposas, que permita no solo evaluar el conocimiento de estudiantes de biología sino también el refuerzo de aprendizaje de un sistema de identificación que utiliza métodos conocidos de

clasificación, a partir de información provista por expertos.

Este proyecto busca servir de base para proyectos futuros orientados al registro de bio-diversidad, permitiendo la inclusión de diferentes áreas de conocimiento como biología, visión por computador e inteligencia artificial. Por las razones expuestas y considerando que este tipo de proyecto se alinea a los intereses del gobierno Ecuatoriano, se considera de importante relevancia.

# CAPÍTULO 1

## 1. Descripción y Justificación

### 1.1 Descripción del problema

La problemática abordada en este trabajo está relacionada al área de biología. Esta disciplina abarca un enfoque específico orientado al estudio y descripción de seres vivos, ya sea como organismos individuales o como especies. Estas especies presentan en algunos casos diferencias significativas entre ellas, que facilitan su identificación, aunque esto no es siempre sencillo cuando se categoriza detalladamente por géneros.

La solución planteada en esta propuesta está orientada de forma específica a una especie en particular, los *lepidópteros* (mariposas), en su última etapa de ciclo de vida. El proceso de recolección de muestras de estas especies implica tiempo y esfuerzo, antes de ser sometidas a estudios para determinar

su evolución y características principales. Expertos en esta área de estudio han logrado generar a lo largo del tiempo, bases de conocimiento sobre familias, sub-familias, tribus y géneros de mariposas con diferentes características relevantes.

Debido a esto, los estudiantes de biología cuentan actualmente con una gran cantidad de información acerca de lepidópteros. Sin embargo, esta cantidad dificulta aprender sobre ellas de forma más efectiva. Por tal razón, este proyecto de materia de graduación busca facilitar el aprendizaje sobre lepidópteros, por medio de una aplicación que facilite no solo la evaluación de conocimientos relacionados con mariposas sino también la colaboración de expertos a nivel mundial en la indexación y clasificación de especímenes, por medio de una herramienta colaborativa.

## **1.2 Justificación del proyecto**

La identificación automática de especies no ha sido un área de interés reciente en biología [1], aunque ha sido de gran interés desde algunas décadas en visión por computador. El estudio realizado en [2] describe un sistema de identificación visual de plantas, que usa una fotografía tomada por el usuario, devolviendo imágenes adicionales de la especie junto con descripciones del mismo, por medio de un dispositivo móvil. En el mundo de los animales acuáticos también, se ha realizado identificación de ballenas y delfines considerando los bordes de sus aletas [3].

Varios estudios han despertado un gran interés por parte de expertos en el área de biología, para encontrar información específica o requerida en la



identificación de algún tipo de espécimen.

Un estudiante de biología que se encuentre analizando o estudiando lepidópteros ya sea mediante la captura de una foto o tomando una muestra si está disponible, anotando todas sus características anatómicas (formas, tamaños, colores, etc. ) y hábitat de la especie, tendrá que enfrentarse a una enorme cantidad de información que deberá ser organizada en grupos y subgrupos con el objetivo de facilitar la disponibilidad de información y reducir el tiempo de recolección de datos.

Dentro de este contexto, algunos estudios han llevado a cabo la identificación automática de patrones por medio de algoritmos de clasificación. En este trabajo, tres algoritmos de clasificación han sido evaluados y seleccionados con métricas como simplicidad, popularidad y eficiencia, tomadas de [4]. Entre los algoritmos más populares se encuentra K-vecinos más próximos (K-NN), este algoritmo ha sido usado en varios trabajos como en [5] donde también se utilizó Máquina de Soporte Vectorial (SVM) dentro del contexto de la identificación de patrones de movimiento. En otro estudio reciente [6] fueron aplicados Perceptron Multicapas (MLP) y SVM.

Para el desarrollo de un sistema colaborativo que permita la identificación de géneros de mariposas, fue abordada una metodología para la extracción de características de imágenes. Posteriormente, se implementó un algoritmo de clasificación (K-NN) para el reconocimiento automático de los patrones resultantes, tomando como referencia el estudio realizado en [5] donde se utilizaron descriptores basados en forma, específicamente, tres momentos invariantes de Hu, con potencias normalizadas.

Adicionalmente, fue desarrollada una aplicación móvil para la visualización de las imágenes de mariposas y la evaluación de conocimientos de estudiantes acerca del género al que pertenecen. La respuesta dada por el estudiante puede ser verificada posteriormente con la ayuda de un experto en el área.

Con el fin de llevar el enfoque propuesto a un contexto más general, este trabajo considera la posibilidad de la adaptación de la metodología planteada a entornos móviles y de bajo poder computacional, lo cual permitiría, por ejemplo, realizar un reconocimiento automático de especies de plantas utilizando un robot móvil o un vehículo radio controlado.

### **1.3 Solución propuesta**

Un modelo general clásico de un sistema automático para la identificación de especímenes, está constituido por las siguientes etapas, el pre procesamiento de las imágenes, selección y extracción de características, y finalmente el uso de un clasificador.

La primera etapa puede ser considerada como el procesamiento inicial para resaltar el objeto de interés. La segunda etapa, un método que permite identificar propiedades relevantes de la imagen pre procesada. La tercera etapa está representada por un módulo para la extracción de patrones y por último, un algoritmo para la clasificación de los patrones extraídos.

El proceso de preparación de las imágenes, extracción y clasificación de características es una parte esencial de la solución planteada. En este trabajo

sólo se mencionan tres técnicas de clasificación más utilizadas en la literatura relacionada. Con ello, se busca identificar especies de mariposas por género, a partir de la información facilitada por el parque nacional *Sangay*.

La Figura 1.1. muestra la estructura del sistema de evaluación de conocimientos sobre especímenes de mariposas.

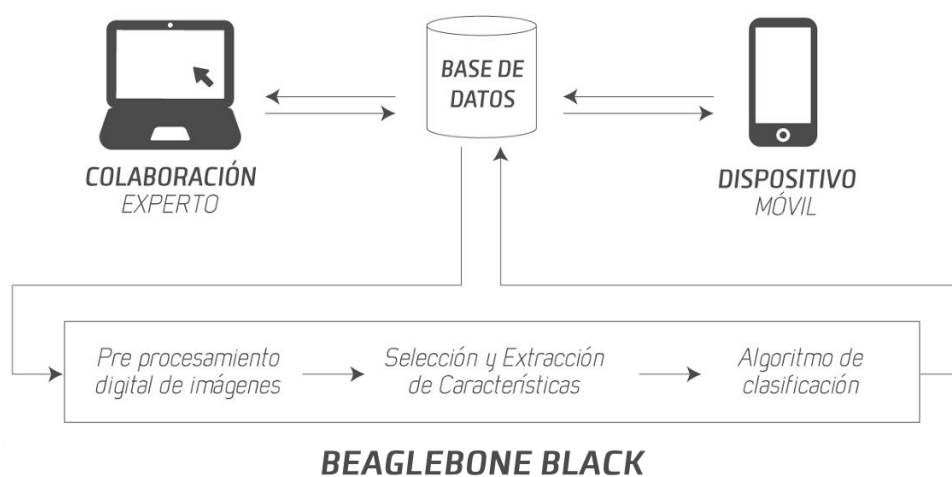


Figura 1.1: Estructura del sistema de evaluación de conocimientos sobre especímenes de mariposas.

## 1.4 Objetivos

### 1.4.1 Objetivo General

Implementar un sistema colaborativo de identificación de especímenes que facilite el aprendizaje de géneros de mariposas.

### 1.4.2 Objetivos Específicos

Los objetivos específicos del presente trabajo son los siguientes:

1. Evaluar tres algoritmos de clasificación de patrones utilizando un conjunto de entrenamiento de 125 especímenes promedio por género.
2. Desarrollar un script para el pre procesamiento de imágenes de mariposas que pertenecen al banco de datos del Parque Nacional Sangay.
3. Implementar un algoritmo para la extracción de características de imágenes basadas en forma, para la generación de patrones de mariposas.
4. Implementar una técnica de clasificación para la identificación automática de ocho géneros de mariposas obtenidas de la reserva ecológica Sangay.
5. Integrar los módulos anteriores en un único sistema, para ser implantado en un dispositivo embebido.
6. Desarrollo de una aplicación para dispositivos móviles (Android) que permita la visualización de imágenes y preguntas sobre especies de mariposas a estudiantes de biología.

## 1.5 Organización del documento

El presente documento se encuentra estructurado de la siguiente manera:

- En el Capítulo 1, se describe la problemática a ser abordada, la justificación de la propuesta, sus objetivos generales y específicos.
- En el Capítulo 2, se presenta un análisis bibliográfico sobre aplicaciones para dispositivos móviles orientados a la educación, especialmente para

el uso en biología, así como también sobre los recursos disponibles para la clasificación de imágenes orientadas al registro automático de biodiversidad.

- El Capítulo 3, detalla los componentes del sistema, la instalación y configuración del Hardware y Software.
- En el Capítulo 4 se detalla la solución propuesta para la identificación automática de mariposas y retroalimentación por parte de los expertos.
- En el capítulo 5 se realiza una evaluación de los resultados obtenidos en el desarrollo del sistema.
- El Capítulo 6, muestra las conclusiones y experiencias durante la implementación del proyecto, así como también los trabajos futuros derivados de esta iniciativa.

## **CAPÍTULO 2**

### **2. Revisión Bibliográfica**

El presente capítulo presenta una revisión del estado del arte relacionado con la temática abordada en este trabajo, junto con la descripción de algunas de las técnicas más comunes utilizadas en trabajos relacionados.

#### **2.1 Pre procesamiento de imágenes**

En algunos trabajos, como en [1], se realiza la identificación de especies de plantas y se menciona una etapa de pre procesamiento de imágenes, aplicada para la segmentación de las hojas de una planta. La imagen es llevada al espacio binario separando la hoja del fondo y convirtiéndola en una imagen binarizada resaltando los bordes de la misma.

Entre los algoritmos de pre procesamiento destinados al reconocimiento óptico

de caracteres(OCR) [7], se menciona que la operación de binarizado cumple un papel fundamental debido a que es el encargado de separar del fondo del documento los rasgos característicos correspondientes a los caracteres.

Para realizar la binarización de la imagen, inicialmente pasamos la imagen por la conversión a escala de grises. La binarización cambia los valores de pixeles de la imagen dependiendo de su umbral de referencia, siendo así que los valores mayores al umbral toman un valor de blanco (255) y los demás toman valor negro (0).

## 2.2 Extracción de Características

Esta sección describe la etapa de extracción de características del conjunto de imágenes de mariposas provistas por la reserva ecológica Sangay. Este proceso permite reducir la cantidad de información de cada uno de los objetos estudiados, obteniendo de esta forma, un vector de características que represente de la mejor manera posible la imagen original [8].

Existen diferentes descriptores de imágenes que pueden ser aplicados, a continuación se mencionan algunos de los métodos más utilizados:

### Método de Descriptores

#### 1. Descriptores de Forma.

Estos descriptores proveen información acerca de las características de forma del objeto. En el estudio [8] se mencionan los siguientes tipos de descriptores basados en formas:

- *Descriptores de Contorno*: Información de pertenencia al

contorno: código de cadenas, firmas, aproximaciones poligonales, descriptores de Fourier, esqueletización, representación polar.

- *Descriptores de Región*: Información de pertenencia al interior del objeto, no sólo con los del contorno: momentos (área, centro de gravedad, momentos invariantes), esqueletos, descriptores topológicos.

## 2. Descriptores de Textura.

Un descriptor de textura analiza la superficie de una imagen para hallar un patrón espacial básico que puede ser periódico ó parcialmente periódico, toma en cuenta la distribución de valores de intensidad del color a nivel espacial, permitiendo evaluar propiedades como fineza, rugosidad, suavidad, granulación, aleatoriedad, entre otros [9].

Estos descriptores deben ser, en la medida de lo posible independientes del tamaño, localización u orientación del objeto para dar mayor flexibilidad al proceso de captura.

### **Momentos invariantes**

Existen varias técnicas para extraer características en fenómenos o objetos de interés, sin embargo estas pueden resultar sensibles a los cambios de posición o escala del objeto. A continuación veremos una técnica citada en la literatura que ha sido de gran utilidad en el presente trabajo, correspondiendo a la categoría de descriptores basados en forma.

### **Momentos Invariantes de Hu**



Los momentos invariantes de un objeto son generados a partir de los momentos centrales normalizados de una imagen en escala de gris. Los momentos centrales son calculados considerando el centroide, en este caso se refiere a un punto en el objeto de interés que describa la misma distancia a los bordes en cualquier dirección.

Los momentos invariantes de  $Hu$  son una combinación no lineal de los momentos centrales normalizados. Considerando  $f(x, y)$  como una imagen digital los momentos centrales pueden ser representados por:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y). \quad (2.1)$$

donde  $\bar{x}$  y  $\bar{y}$  son los componentes del centroide [5]. La invariancia de escala es obtenida a partir de los momentos centrales y se determina por:

$$\eta_{ij} = \frac{\mu_{ij}}{u_{00}^{(1 + \frac{i+j}{2})}}, \quad (2.2)$$

A partir de los momentos centrales normalizados es posible extraer los momentos de  $Hu$  de orden dos, tres y cuatro utilizados en este trabajo. Estos tres momentos de  $Hu$  fueron determinados en el estudio [5] como los más eficientes y de los mejores en cuanto a su ortogonalidad, determinados por:

$$\begin{aligned} Hu_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ Hu_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ Hu_4 &= (\eta_{30} - \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \end{aligned} \quad (2.3)$$

Una ventaja de los momentos invariantes es la simplicidad del algoritmo, sobre

todo si se lo aplica en imágenes binarias [8].

## 2.3 Algoritmos de Clasificación

### Tipos de Aprendizaje

Existen dos enfoques principales para el aprendizaje artificial, aprendizaje supervisado y aprendizaje no supervisado.

- **Aprendizaje Supervisado**, este tipo de aprendizaje trata de minimizar el error de la salida calculada respecto a la salida deseada, es decir, trata de corregir o modificar los pesos en las conexiones de la red para que su salida coincida con las salidas deseadas. Para esto se necesita un conjunto de datos de entrada previamente conocidos.

En el aprendizaje supervisado el entrenamiento consiste en presentarle a la red patrones de entrada pertenecientes a un conjunto de ensayo “patrón de estímulos - respuesta correcta” que debe ser elegido cuidadosamente para que pueda representar equilibradamente toda la información que la red necesita aprender [10].

- **Aprendizaje No Supervisado**, para este aprendizaje no se necesita un conjunto de datos de entrada previamente conocidos, debido a que se espera extraer rasgos o agrupar patrones similares por familiaridad o asociaciones entre los patrones que ocurren frecuentemente. Una vez que ellos hayan aprendido su asociación, le permitirá funcionar en modo de recuerdo.

## Redes Neuronales Artificiales

Una neurona biológica está formada por un cuerpo que contiene entradas denominadas dendritas y un axón que comprende la salida. Existen conexiones entre neuronas, a esto se lo denomina sinapsis, lo cual permite que se almacene el conocimiento. Estas conexiones se dan entre capas o en la misma capa que contiene a un grupo de neuronas.

Una neurona artificial tiene una infraestructura inspirada en una neurona biológica, como lo vemos en la Figura 2.1. Pero en términos de escala un cerebro contiene un número de neuronas mucho mayor que cualquier red neuronal artificial creada.

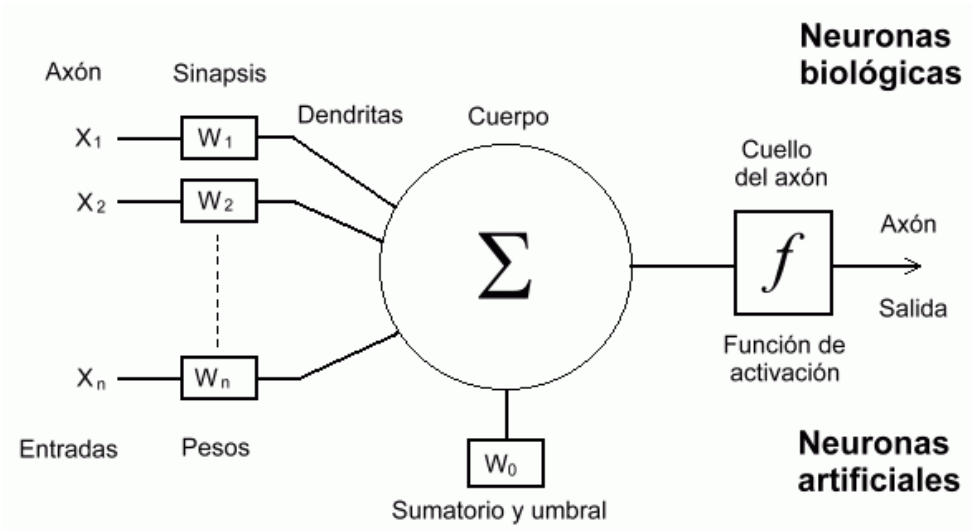


Figura 2.1: Similitud entre neurona biológica y artificial.

El cerebro logra el razonamiento por medio de conexiones sinápticas entre las neuronas, de la misma manera las RNA (Redes Neuronales Artificiales) reproducen un comportamiento similar al de las neuronas biológicas. Estas conexiones se establecen mediante un modelo matemático.

Las diferentes arquitecturas de redes neuronales artificiales tienen distintas propiedades dependiendo de la manera en que sus componentes se interconectan y comparten información.

Podemos mencionar varios modelos de redes neuronales que han sido utilizados, un caso reciente es mostrado en [11], donde se utilizaron redes neuronales artificiales como: redes neuronales recurrentes (RNN), redes neuronales feedforward (FF) con algoritmos genéticos y el algoritmo LM (Levenberg–Marquardt) en la clasificación de especímenes de diferentes té de frutas con nuevos sistemas híbridos.

Diferentes propuestas para la mejora de modelos de aprendizaje han sido probados entre enfoques estadísticos [12] y métodos de clasificación basados en RNA [13]. Esta área de investigación ha llevado a muchos a mejorar el diseño de topologías de estas redes, aunque como cita el estudio en [14] no hay manera sistemática para diseñar una arquitectura óptima para una tarea determinada. Sin embargo, estos avances han tenido muy buenos resultados en los planteamientos de problemas según su naturaleza.

En general, todos los sistemas basados en redes neuronales son diseñados con el fin de generalizar y en ciertos casos predecir patrones o estados deseados a través del entrenamiento y prueba de dichos sistemas.

Entre los numerosos algoritmos que existen para problemas de clasificación, se han escogido tres métodos de clasificación: K-vecinos más cercanos (KNN), Perceptrón Multicapa (MLP) y Máquinas de Soporte Vectorial (SVM) dentro del

paradigma "Aprendizaje Supervisado", considerando los estudios realizados en [5].

### 2.3.1 K-vecinos más cercanos (KNN)

KNN se basa en la búsqueda de los k prototipos más cercanos al patrón a clasificar, dentro de un conjunto de prototipos de entrenamiento.

Existe dos etapas dentro de la implementación de este algoritmo: fase de entrenamiento y fase de clasificación. La fase entrenamiento almacena las etiquetas y los vectores de entrenamiento. Para la clasificación se presenta el vector de características cuya clase es desconocida, luego se procede a realizar un cálculo de distancia entre los k vectores característicos más cercanos, por lo general se utiliza la distancia Euclidiana que está dada por la siguiente ecuación:

$$d_E (P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.4)$$

Obteniendo la distancia entre el vector característico y los vectores de entrenamiento se selecciona los K ejemplos más cercanos del espacio de características.

El proyecto descrito en [10], planteó el uso de este algoritmo dentro el aprendizaje computacional para el bloqueo automático de páginas web que incitan a la violencia, indicando ser un modelo efectivo de clasificación.

### 2.3.2 Perceptrón Multicapa (MLP)

MLP es un modelo de alimentación "hacia adelante" que mapea conjuntos de datos de entrada en una serie de conjuntos de datos de salida [13]. Es la función de activación la que mapea las entradas de cada neurona a su salida. Este modelo a diferencia de otras redes neuronales maneja una función no lineal, siendo la función sigmoide generalmente usada para este propósito y viene dada por la siguiente ecuación:

$$P(t) = \frac{1}{1 + e^{-t}} \quad (2.5)$$

Esta arquitectura comprende una capa de entrada, con una o varias capas ocultas y una capa de salida, dentro de estas capas las neuronas se encuentran conectadas con otras neuronas de capas previas y siguientes. Estas redes pueden comportarse como clasificadores y también como aproximadores de funciones [6]. Una ventaja de este modelo es que se pueden distinguir los datos que no son linealmente separables [13].

Varios trabajos se han desarrollado bajo la evaluación del algoritmo Perceptrón Multicapa (MLP), un estudio descrito en [6] logra obtener buenos resultados de clasificación en el reconocimiento de caracteres manuscritos, utilizando esta arquitectura como una herramienta adecuada.

### 2.3.3 Máquina de Soporte Vectorial (SVM)

Es un algoritmo para aprendizaje automático que busca maximizar la separación entre clases por medio de vectores de soporte. Representa una de las técnicas más utilizadas para aplicaciones como: reconocimiento de rostros, categorización textual, predicciones, recuperaciones de imágenes y reconocimiento de escritura manuscrita. Es muy utilizado por su excelente capacidad de generalización aún en espacios de alta dimensión y utilizando conjuntos de entrenamiento pequeños [6].

Originalmente se diseñó como un clasificador binario, éste usa una función núcleo que mapea los datos a un espacio de mayor dimensión, con el objetivo de mejorar la clasificación. La función núcleo está definida por:

$$y_i = \begin{cases} 1 & \text{si } x_i \text{ está en la clase } 1, o \\ -1 & \text{si } x_i \text{ está en la clase } 2 \end{cases} \quad (2.6)$$

Dados los vectores de entrenamiento  $x_i$ ,  $i = 1, \dots, l$  de longitud  $n$ , y el vector  $y_i$  definido por la ecuación 2.6. Esta técnica busca obtener el mayor margen de separación entre dos clases, medido desde los vectores de soporte hacia el hiperplano [6].

## **CAPÍTULO 3**

# **3. Instalación y Configuración**

### **3.1 Introducción**

En este capítulo se detallan las etapas de instalación y configuración necesarias para el funcionamiento del sistema. Al inicio de este documento se describe el propósito de los módulos que conforman este proyecto, uno de ellos es utilizar una herramienta de bajo costo para la implementación de un sistema colaborativo.

A continuación serán detallados los procedimientos para la instalación y configuración de la solución planteada en este trabajo.



## 3.2 Hardware/Software

Las herramientas para la implementación de nuestro sistema están conformadas por:

- BeagleBone modelo Black.- Es una computadora de bajo costo que fue diseñada para la integración de hardware y software realizado por desarrolladores familiarizados con lenguajes de programación como Python, C++ y Java, con el interés de aprender y crear nuevas aplicaciones sobre sistemas embebidos.

Compatible con las plataformas GNU Linux y Android, viene por defecto instalado el sistema operativo Angstrom de Linux. Trae consigo un procesador ARM Cortex A8, con una frecuencia máxima de 1.2 GHz. Usa una extensión que sirve para la interconexión del BeagleBone con un dispositivo de preferencia. Contiene una Memoria ultra SD 8GB que sirve para el almacenamiento de información, además tiene una velocidad de lectura de 30 Mb/s.

Varios sistemas han sido desarrollados utilizando plataformas de bajo coste, lo cual permite sean fácilmente configurables y portátiles como lo describe el sistema que fue implementado en [7], diseñado para correr sobre plataformas embebidas como BeagleBone.

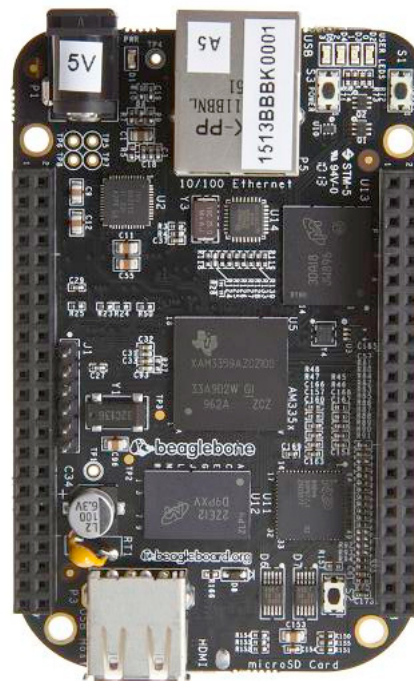


Figura 3.1: BeagleBone Black.

- Samsung I9300 Galaxy SIII.- Dispositivo móvil con sistema operativo Android que permite ejecutar la aplicación desarrollada en este trabajo.



Figura 3.2: Dispositivo móvil Samsung SIII.

- GNU/Linux.- Sistema operativo libre para la ejecución de programas y administración de recursos del computador. Linux fue diseñado como un

sistema multi tarea y multi usuario basado en el sistema operativo Unix.

- Distribución Linux Mint 17 Qiana.- La distribución Mint es utilizada por muchos usuarios debido a su interfaz amigable y similar a Microsoft Windows.
- OpenCV versión 2.4.10.- Es una de las librerías utilizadas para el tratamiento de imágenes. Posee un número considerable de funciones y entre ellas funciones para el análisis basado en la forma de una imagen (contornos, bordes), análisis del movimiento, detección de rasgos, técnicas de calibración (cámara), segmentación de objetos y reconocimiento. Con estas prestaciones, OpenCV permite desarrollar poderosas aplicaciones de visión por computador para el uso académico o comercial. Además tiene soporte para lenguajes de programación comunes como: Python, Java, C, C++ y es compatible con los sistemas operativos: Windows, Linux, Mac OS, iOS y Android.



Figura 3.3: Logo de OpenCV.

- Android Studio.- Es una plataforma utilizada para el desarrollo de aplicaciones en Android, es de fácil uso y permite el diseño rápido de aplicaciones.
- Symfony 2.3.- Framework consumido por muchos programadores para crear aplicaciones y sitios web. Trabaja con una plantilla llamada Twig y permite incorporar diferentes funcionalidades en un proyecto mediante el uso de sus componentes. Tiene soporte y muy buena documentación.

### 3.2.1 Instalación del Hardware

Para la implementación del sistema, iniciamos con la preparación de la tarjeta BeagleBone Black. Esta placa se conecta con un cable USB a un computador, instalando todos los controladores necesarios. Esta placa tiene preinstalado el sistema operativo Linux Angstrom, pero se decidió instalar el sistema operativo Linux/Debian por su mejor rendimiento (baja memoria para trabajar) y compatibilidad con la librería OpenCV, además de poseer una interfaz amigable.

La tarjeta trae consigo una memoria SD, en esta memoria estarán almacenados los archivos correspondientes al sistema operativo, estos recursos pueden ser encontrados en la página principal de BeagleBone. Luego de la instalación del sistema operativo procedemos a realizar la configuración de nuestro sistema. Para conectarse con el sistema instalado procedemos a realizar los siguientes pasos:

1. Para tener acceso remoto al sistema instalamos un programa llamado VNC (Computación Virtual en Red) el cual permitió conectarnos remotamente desde cualquier ordenador al dispositivo embebido, el ordenador-servidor es nuestra placa BeagleBone. Para instalar VNC ejecutamos en consola el siguiente comando:  
**apt-get install x11vnc**
2. Procedemos a conectarnos mediante SSH (Secure Shell), un intérprete de órdenes seguras para acceder a máquinas remotas a través de una red. Existe una dirección IP asignada a la interfaz de red de la placa BeagleBone, copiamos esta IP y procedemos a conectarnos mediante **ssh debian@IP**.

3. Necesitamos ejecutar el servicio VNC desde la placa, para que pueda establecer la conexión con el paso anterior. Corremos el siguiente comando por consola:

```
x11vnc -bg -o %HOME/.x11vnc.log.%VNCDISPLAY -auth  
/var/run/gdm/auth-for-gdm*/database -display :0
```

### 3.2.2 Instalación de Software

Para la instalación de algunas librerías utilizadas en este estudio, encontramos primero la librería OpenCV la cual se procedió de la siguiente manera:

1. Instalar las dependencias de software, para esto utilizamos el siguiente comando: **sudo apt-get install built-essential**
2. Descargar la última versión de OpenCV estable disponible en la página oficial.
3. Abrir el directorio de OpenCV que fue descargado mediante el siguiente comando:

```
cd /opencv
```

Luego de esto se crea una carpeta y al ingresar a esta, ejecutamos el comando:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D  
CMAKE_INSTALL_PREFIX=usrlocal ..
```

4. Luego de realizar el paso anterior procedemos con los siguientes comandos:

**make**

**sudo make install**

Otra librería usada en este trabajo es cppnet-lib desarrollada en C++ la cual permite la conexión desde el sistema de identificación de mariposas al servidor, donde almacena la información generada por el mismo. Se encuentra disponible la última versión y documentación en la página oficial.

En el caso de Android Studio se procede a descargar el SDK (Kit de Desarrollo de Software) desde la página oficial, dentro de la carpeta del SDK encontraremos el script studio.sh, que al ser ejecutado levantará el IDE (Ambiente de Interfaz de Desarrollo) para aplicaciones móviles.

### 3.2.3 Arquitectura básica del sistema

La Figura 3.8 muestra la arquitectura básica del sistema para evaluación de especímenes. Está conformada por los componentes presentados a continuación:

El **sistema embebido** extrae desde un directorio el conjunto de mariposas de entrenamiento y posteriormente el conjunto de evaluación con el cual realiza la categorización de las mariposas de acuerdo al género. Luego se conecta al servidor enviando la información categorizada de las mariposas a la base de datos.

El dispositivo que se encuentra ejecutando la aplicación móvil del sistema muestra al usuario los cuestionarios o evaluaciones a realizar después de consultarlos en la base de datos. Posterior a la evaluación

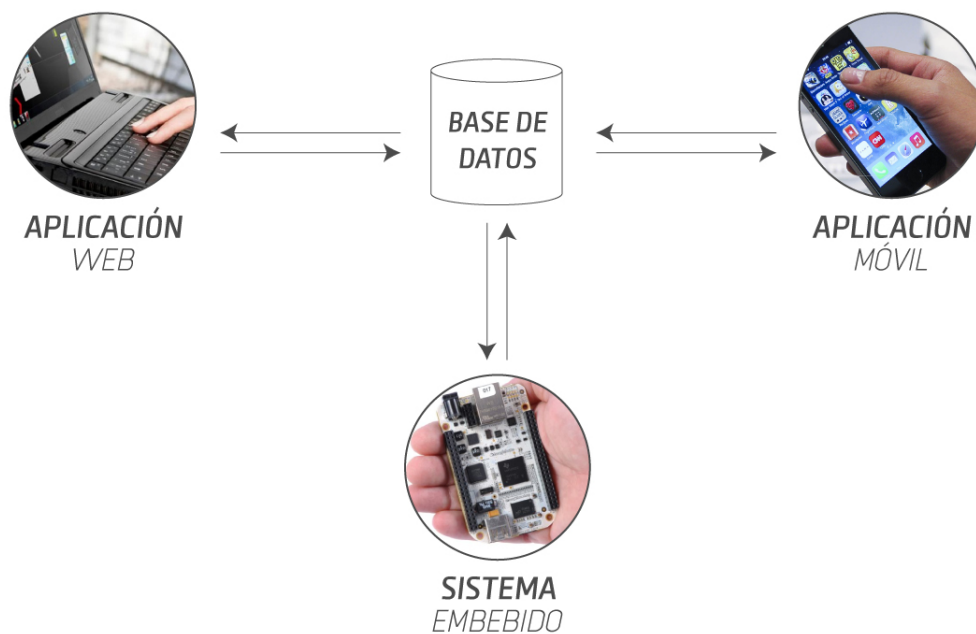


Figura 3.4: Arquitectura básica del sistema

realizada por el estudiante, sus respuestas se alojan nuevamente en la base de datos esperando una verificación por parte del experto.

Por medio de una aplicación web el experto corroborará las respuestas de cada estudiante, además de contribuir al conjunto de entrenamiento.

## **CAPÍTULO 4**

# **4. Desarrollo y Implementación.**

### **4.1 Introducción**

La solución desarrollada está compuesta por dos componentes distintos dentro de este trabajo. El primero es el desarrollo de un sistema que permita realizar la identificación de mariposas por género. El segundo, el desarrollo de una aplicación móvil que permita visualizar los resultados del primer componente.

### **4.2 Desarrollo del Sistema**

La primera fase consiste en el desarrollo del sistema para el dispositivo embebido, que comprende el clasificador de especímenes. El sistema se encuentra instalado en la placa "BeagleBone Black". Para el desarrollo del sistema identificador de mariposas procedemos de la siguiente manera:



### **4.2.1 Conversión a escala de grises**

En esta etapa se procede a ejecutar un programa realizado en shell script para procesar las imágenes de entrenamiento y convertirlas automáticamente en escala de grises, ya que el algoritmo de clasificación utilizado está diseñado para trabajar en un solo canal. Se debe realizar este proceso previo porque las imágenes provistas por la reserva ecológica Sangay se encuentran disponibles a color.

Este proceso de conversión a escala de grises es uno de los primeros pasos para el desarrollo de la solución planteada en este trabajo.

### **4.2.2 Binarización**

Luego se procede a convertir la imagen recibida en una imagen binaria separando así el fondo de la especie a analizar. Al obtener solamente la forma de la imagen, se obtienen características que corresponden exclusivamente al objeto de interés.

### **4.2.3 Extracción de características**

Para esta etapa se utilizan los momentos invariantes de Hu considerando la diferencia de forma que existen entre los diferentes géneros de mariposas.

Ésta técnica permite generar un vector de tres elementos, que contiene las características relevantes de cada imagen. Este proceso extrae representaciones numéricas del objeto de interés para ser utilizado en la entrada del algoritmo.

#### **4.2.4 Algoritmo de clasificación de mariposas**

Se evaluaron los algoritmos KNN (K vecinos más cercanos), SVM (Máquina de Soporte Vectorial) y MLP (Percéptron Multicapas) planteados al inicio de la revisión bibliográfica. Esta experimentación fue llevada a cabo con un número de 125 imágenes aproximadamente por género abarcando un total de ocho géneros a categorizar, esta información fue utilizada para el conjunto de entrenamiento del sistema.

Estos algoritmos mantuvieron su desempeño dentro de un margen similar entre ellos, aunque los mejores resultados se obtuvieron con SVM y KNN, sin embargo, la implementación de K-NN es más sencilla y por razones de tiempo fue la opción escogida para este trabajo. El uso del algoritmo KNN permite conocer el número de patrones más cercano al patrón evaluado, este número está descrito por K, lo cual fue establecido en un valor igual a 3, puesto que una de las características del sistema de identificación de mariposas es presentarle al estudiante tres opciones o géneros posibles a evaluar.

### **4.3 Desarrollo del Aplicativo Móvil**

Basado en las aplicaciones existentes se consiguió diseñar una interfaz gráfica de fácil uso para el usuario.

#### **4.3.1 Introducción**

A partir del diseño de la aplicación móvil se procedió a buscar una herramienta o IDE para desarrollar el aplicativo IDM (Identificación de Mariposas), se utilizó "Android Studio" como framework de desarrollo

usando el lenguaje de programación Java. En la Figura 4.1 se muestra el ambiente de desarrollo utilizando Android Studio.

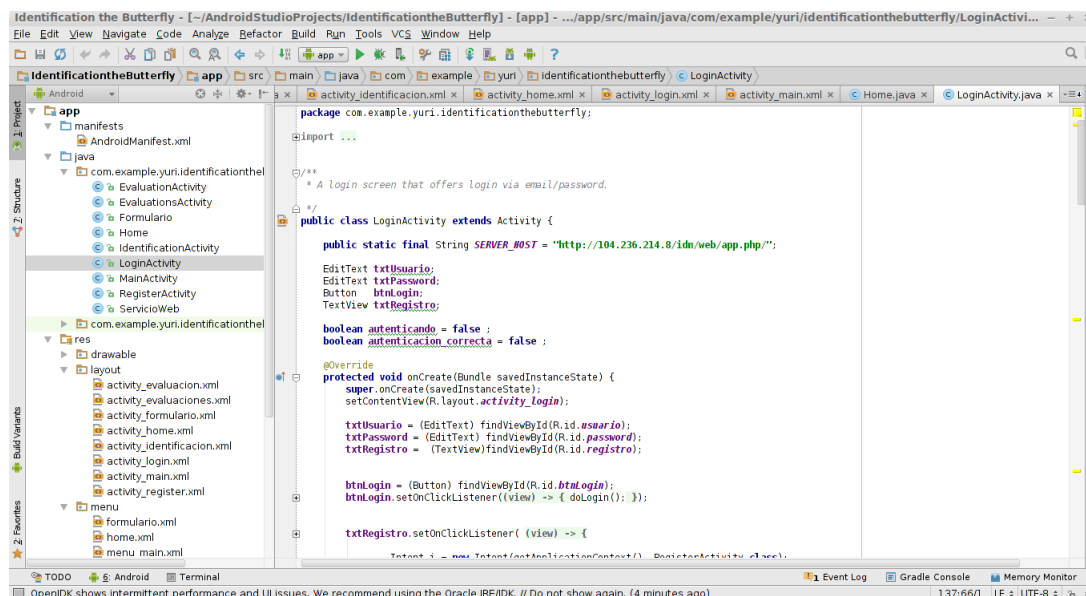


Figura 4.1: Desarrollo en Android Studio.

Android Studio simula un dispositivo móvil, donde podemos tener una vista previa de nuestra aplicación, Figura 4.2.

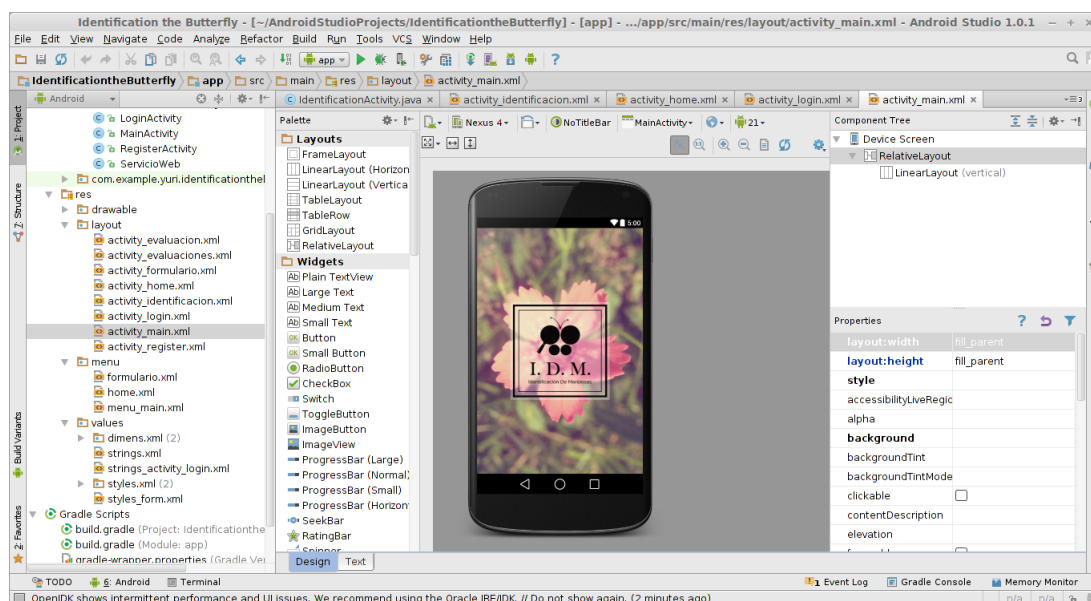


Figura 4.2: Aplicativo en Android Studio.

### 4.3.2 Análisis de Requerimientos y Diseño de la plataforma móvil

A continuación una descripción general de los requerimientos en la plataforma móvil.

- Permitir el ingreso de los usuarios que van a tener acceso al aplicativo móvil con sus respectivos permisos.
- Registrar usuarios como estudiantes para hacer uso de la aplicación.
- Registrar las evaluaciones que son realizadas por los estudiantes.
- Mostrar los resultados de las evaluaciones pertenecientes al estudiante.
- Permitir el ingreso de usuarios expertos para registrar su retroalimentación sobre las evaluaciones realizadas a los estudiantes.

Para el diseño de la aplicación se elaboró una interfaz de fácil uso y amigable con el usuario. A continuación se muestran unas capturas de pantallas del aplicativo móvil.

| Pantalla  | Descripción                                    |
|---|--|
|    | Pantalla de inicio de la aplicación.           |
|   | Pantalla de nueva evaluación o identificación. |
|  | Pantalla de visualización de resultados.       |

Tabla 4.1: Pantallas del aplicativo móvil.

#### 4.4 Colaboración Alumno / Experto

El sistema web ha sido desarrollado en PHP con la ayuda del framework *Symfony*, una herramienta útil que provee varios componentes entre ellos el

*bundle sonata project* que genera una pantalla de administración amigable. Este constituye un vínculo entre el estudiante y el experto en el área de reconocimiento de lepidópteros, con el objetivo de validar las respuestas del estudiante y las respuestas del sistema de identificación de mariposas.

Este sistema administrativo provee una interfaz gráfica vía web, creada específicamente para el administrador del sistema y expertos. En la Figura 4.3 se presenta un formulario de ingreso para el experto ya registrado por el administrador, en este caso su usuario y contraseña sería la palabra "admin" para ambos.

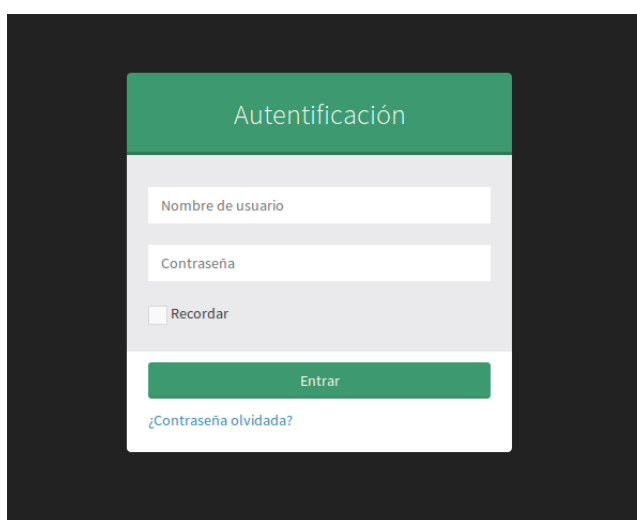
The image shows a web login form with a green header containing the word "Autenticación". Below the header, there are two input fields: "Nombre de usuario" and "Contraseña". Underneath these fields is a checkbox labeled "Recordar". At the bottom of the form is a green button labeled "Entrar". Below the button, there is a blue link that says "¿Contraseña olvidada?".

Figura 4.3: Pantalla de ingreso al sistema web.

Además permite visualizar la base de datos de los especímenes que han sido previamente identificados, en la Figura 4.4 se visualiza un ejemplo de una especie identificada por el sistema.

| Id | Nombre       | Usuario | Imagen       | Opcion1 | Opcion2 | Opcion3   | Opcion Correcta | Opcion Seleccionada | Estado Calificacion | Estado    |
|----|--------------|---------|--------------|---------|---------|-----------|-----------------|---------------------|---------------------|-----------|
| 1  | Evaluación 1 | Yuri    | mariposa.png | Adelpha | Della   | Euselasia | Euselasia       | Adelpha             | Verificada          | Publicado |

Figura 4.4: Pantalla del listado de especímenes categorizados.

En la Figura 4.5 podemos observar la pantalla de bienvenida al sistema web con los módulos generados previamente, el sistema también le permite registrar nuevos usuarios para que puedan ingresar y colaborar con su conocimiento en evaluaciones existentes hechas por los estudiantes.

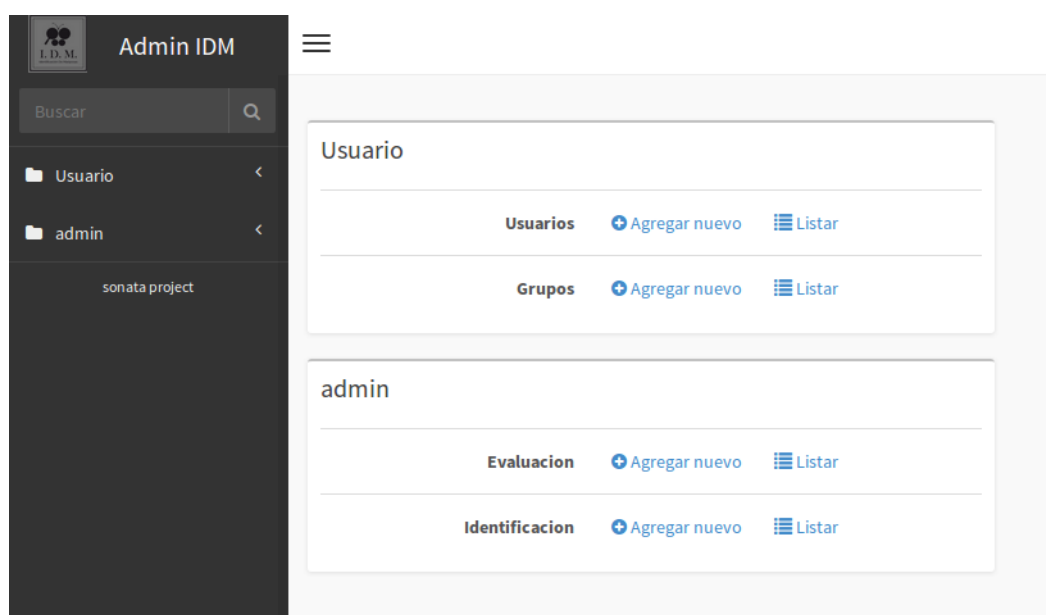


Figura 4.5: Pantalla de inicio al sistema web.

En la Figura 4.6, podemos observar el listado de las evaluaciones hechas por los estudiantes, enviados desde la aplicación móvil. En este módulo se pueden

visualizar, editar y eliminar dichos registros.

| Id | Nombre       | Usuario | Imagen       | Opcion1 | Opcion2 | Opcion3   | Opcion Correcta | Opcion Seleccionada | Estado Calificacion | Estado    |
|----|--------------|---------|--------------|---------|---------|-----------|-----------------|---------------------|---------------------|-----------|
| 1  | Evaluación 1 | Yuri    | mariposa.png | Adelpha | Della   | Euselasia | Euselasia       | Adelpha             | Verificada          | Publicado |

Figura 4.6: Pantalla del listado de evaluaciones.

## 4.5 Implementación del Software

Una vez desarrollados los componentes de este trabajo se procedió a la implementación de las partes. Se muestra en la Figura 4.7 el desarrollo de la implementación del sistema.

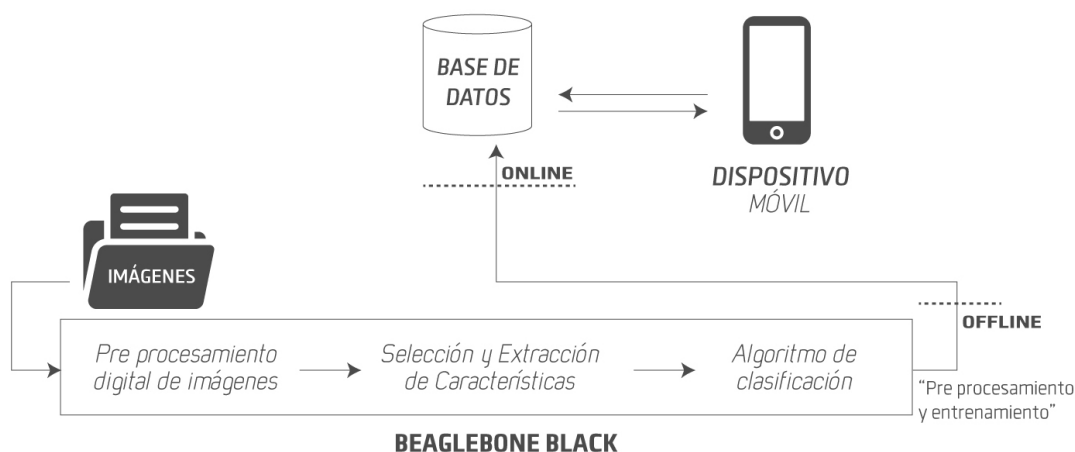


Figura 4.7: Desarrollo de la Impementación del sistema.

El componente desarrollado en el BeagleBone se lo realiza bajo un proceso "offline" como forma de testear el desempeño de los algoritmos de clasificación empleados en este trabajo. Posterior al proceso de entrenamiento, durante



la ejecución del programa se extraen una a una las imágenes del conjunto de evaluación. Finalmente se almacena la información categorizada en la base de datos. Una vez que esta información esté en la base de datos, la aplicación móvil la consultará para el funcionamiento del sistema colaborativo de identificación de especímenes lepidópteros.

Por otro lado, la aplicación móvil, consulta a la base de datos si existen evaluaciones disponibles para el estudiante registrado, luego presenta una imagen en la sección de "Nueva Evaluación" al estudiante, al realizarla es registrada en el servidor web con un estado de "revisión pendiente". Además el estudiante podrá visualizar dicha evaluación en la sección "Evaluaciones" que contiene su historial de evaluaciones y retroalimentación por parte del experto.

Realizado esto, el experto podrá visualizar las evaluaciones en estado de revisión pendiente para corroborar la respuesta generada por el sistema de identificación de mariposas y la respuesta del estudiante como correcta o no. Esto actualizará el estado de las evaluaciones realizadas por el estudiante, proporcionándole retroalimentación de su evaluación.

# **CAPÍTULO 5**

## **5. Análisis de Resultados**

### **5.1 Resultados**

Con la metodología planteada obtuvimos resultados en cada uno de los componentes desarrollados, mostrados mediante gráficos.

#### **1. Pre procesamiento de imágenes**

En esta etapa se implementó un algoritmo hecho para Linux, que realiza el pre procesamiento de las imágenes de mariposas a partir del conjunto de entrenamiento. Al inicio se carga la imagen original, como vemos en la Figura 5.1.

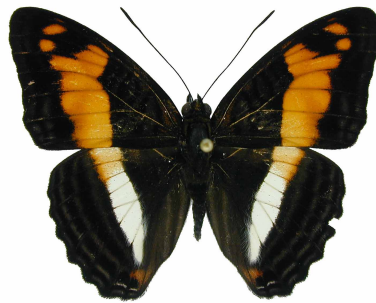


Figura 5.1: Imagen original.

Luego, las imágenes anteriores son llevadas al proceso de conversión a escala de grises, como muestra en la Figura 5.2.



Figura 5.2: Imagen en escala de grises.

Finalmente, realiza el proceso de binarización del conjunto de imágenes, con el objetivo de favorecer el proceso de clasificación, como muestra la Figura 5.3.

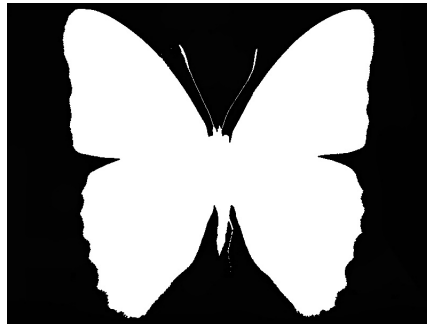


Figura 5.3: Imagen binarizada.

## 2. Método de clasificación

En la Figura 5.4 se visualizan los patrones del conjunto de entrenamiento de mariposas en un plano 3D (tres dimensiones), considerando ocho géneros de mariposas con un total de muestras igual a 125 imágenes por género. En este gráfico se muestran los patrones correspondientes a las mariposas del conjunto de entrenamiento con características de momentos invariantes de Hu (Hu2, Hu3, Hu4).

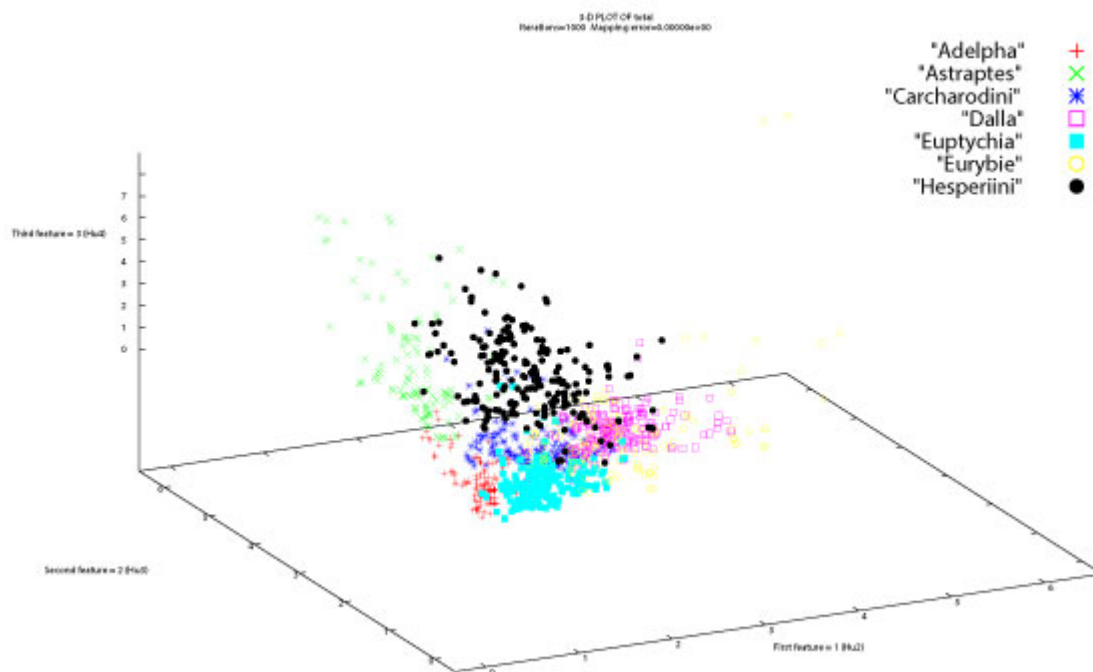


Figura 5.4: Visualización de patrones de mariposas.

En la Figura 5.5 se muestra el conjunto de entrenamiento de mariposas con los patrones mapeados con la ayuda de un Kernel T-Student, para la mejor visualización de la agrupación de los patrones. En un análisis más amplio a ser realizado en un futuro trabajo, el uso de un kernel podría contribuir a la clasificación.

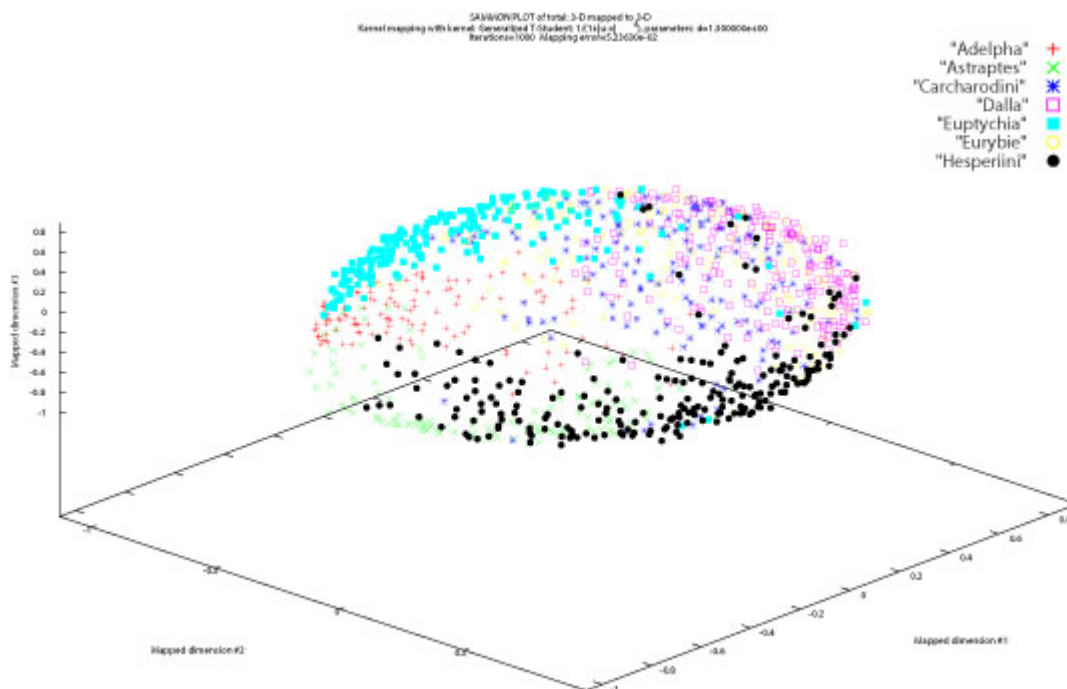


Figura 5.5: Visualización de patrones de mariposas, utilizando kernel T-Student.

De acuerdo a los algoritmos de clasificación mencionados en la revisión bibliográfica, se realizaron algunas pruebas variando el número de géneros del conjunto de entrenamiento, considerando en todos los casos 75% para entrenamiento y 25% para evaluación, desde 2 hasta 8 clases.

- **Test I:** 2 clases: Se obtuvo una precisión del 98% con KNN y SVM, mientras que con MLP el 97%.
- **Test II:** 5 clases: Logró en el peor de los casos el 85.5% con KNN y MLP, y el SVM con 88%.
- **Test III:** 8 clases: SVM logró clasificar con un 75.5% de certeza, siendo que con KNN y MLP se consiguió 72% y 68% respectivamente.

Con la ayuda de *GNU plot* se muestran los resultados de las pruebas realizadas durante la evaluación de los algoritmos antes mencionados, siendo evidente un rendimiento similar utilizando los tres algoritmos de clasificación.

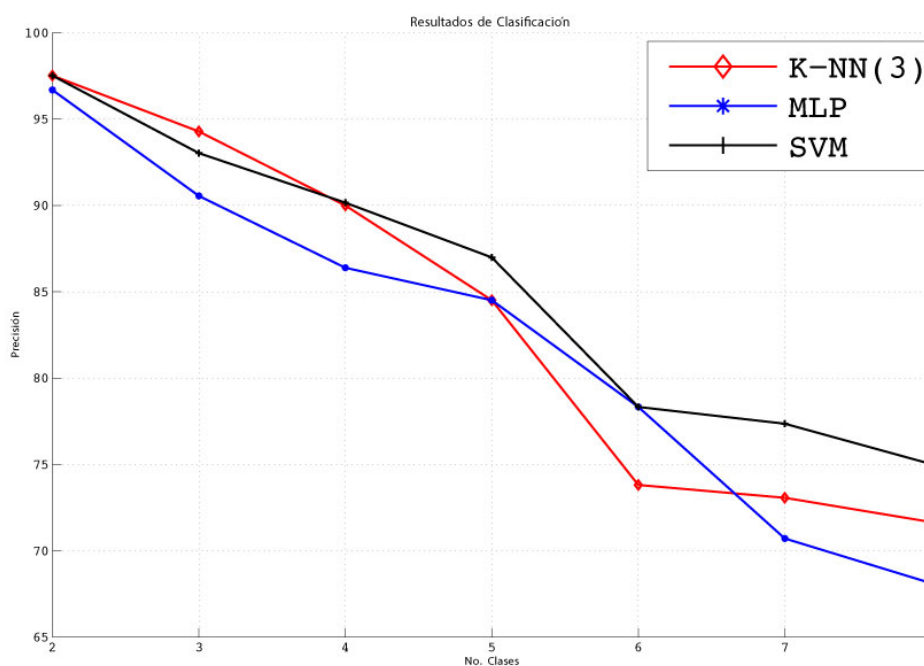


Figura 5.6: Resultados de los algoritmos de clasificación.

Sin embargo, podemos apreciar que el algoritmo que alcanza un porcentaje superior de precisión es SVM aunque manteniéndose cercano a los otros.

### 3. Tiempo de extracción y clasificación de patrones.

La Figura 5.7 muestra una comparación entre un computador actual promedio (procesador de 2 GHz) y BeagleBone Black (1.2 GHz).

|   | Funciones              | PC                 | BeagleBone Black   |
|---|------------------------|--------------------|--------------------|
| <b>Conjunto de Entrenamiento (943 imágenes)</b> | Extracción de Patrones | 168 (ms) /imagen   | 199 (ms) /imagen   |
| <b>Tiempo Total de Entrenamiento</b>            |                        | <b>1':19 (s) ~</b> | <b>3':30 (s) ~</b> |
| <b>Conjunto de Test (208 imágenes)</b>          | Extracción de Patrones | 33 (ms) /imagen    | 437 (ms) /imagen   |
|   | Predicción             | 3 (ms) / imagen    | 54 (ms) / imagen   |
| <b>Tiempo Total de Test</b>                     |                        | <b>6 (s) ~</b>     | <b>1':26 (s) ~</b> |

Figura 5.7: Tiempos de extracción y clasificación de patrones.

Considerando un conjunto de 943 imágenes, el tiempo total de extracción de patrones fue de 1 minuto y 19 segundos, en un computador con 2 GHz de procesador, mientras en un dispositivo embebido BeagleBone Black (1.2 GHz) fue de 3 minutos con 30 segundos aproximadamente.

Considerando un conjunto de prueba de 208 imágenes, posterior al proceso de entrenamiento, se realiza la predicción de la clase a la que pertenece cada una de las mariposas presentadas al clasificador. El proceso de predicción de todo el conjunto de prueba tardó 6 segundos en un computador con 2 GHz mientras que en BeagleBone Black demoró 1 minuto con 26 segundos.

A pesar de las limitaciones del hardware de BeagleBone podemos observar que es posible ejecutar aplicaciones que requieren de clasificación de información con tiempos de respuesta aceptables, lo cual



extiende el alcance de estas aplicaciones a soluciones que aprovechen la portabilidad de un sistema de cómputo de tamaño reducido como el utilizado en este trabajo.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

1. Con el uso de la técnica de momentos invariantes de Hu se obtuvo un vector de tres elementos: Hu2, Hu3 y Hu4 con potencias normalizadas, que permitió alcanzar una precisión del 78% con SVM, 74% con K-NN y el 71% con MLP, considerando 7 géneros de mariposas.
2. Los momentos invariantes de Hu a pesar de ser una técnica utilizada desde años atrás son rápidos de calcular, esta ventaja tiene un peso fundamental dado las limitaciones de hardware planteadas inicialmente.

### Recomendaciones

1. Es importante considerar la calidad del conjunto de entrenamiento. Imágenes incompletas o dañadas perjudican la homogeneidad del conjunto de datos y por consiguiente la clasificación.
2. La etapa de pre procesamiento de imágenes es relevante y se recomienda realizar una binarización que permita resaltar la forma del objeto y obtener así mejores características.

## **Trabajos Futuros**

Como trabajo futuro se pretende incorporar SVM para la clasificación de mariposas y ampliar los estudios relacionados al proceso de extracción de características para mejorar los resultados del proceso de clasificación con un número mayor de clases.

En la aplicación móvil se podría incorporar una guía acerca de la información básica de los especímenes mariposas.

## BIBLIOGRAFÍA

- [1] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012*, pages 502–516. Springer, 2012.
- [2] Peter N Belhumeur, Daozheng Chen, Steven Feiner, David W Jacobs, W John Kress, Haibin Ling, Ida Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, et al. Searching the world's herbaria: A system for visual identification of plant species. In *Computer Vision–ECCV 2008*, pages 116–129. Springer, 2008.
- [3] Scott A Hale. Unsupervised threshold for automatic extraction of dolphin dorsal fin outlines from digital photographs in darwin (digital analysis and recognition of whale images on a network). *arXiv preprint arXiv:1202.4107*, 2012.
- [4] María Guijarro Mata-García, Gonzalo Pajares Martinsanz, and Pedro Javier Herrera Caro. Diseño de un nuevo clasificador supervisado para minería de datos. *Anales de ingeniería técnica en informática de sistemas*, (3):112–128, 2010.
- [5] Dennis Romero López, Anselmo Frizera Neto, and Teodiano Freire Bastos. Reconocimiento en-línea de acciones humanas basado en patrones de rwe aplicado en ventanas dinámicas de momentos invariantes. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 11(2):202–211, 2014.
- [6] Leticia María Seijas. Reconocimiento de patrones utilizando técnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos.
- [7] Claudio Delrieux, Ezequiel Pawelko, Leandro Arcusin, and Marcelo Raponi. Procesamiento de imágenes en tiempo real utilizando tecnología embebida. In *XVIII Congreso Argentino de Ciencias de la Computación*, 2012.
- [8] Eddie Angel Sobrado Malpartida. Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. 2011.
- [9] Nora La Serna Palomino, Walter Contreras Flores, and María Elena Ruiz Rivera. Procesamiento digital de textura: Técnicas utilizadas en aplicaciones actuales de cbir. *Revista de investigación de Sistemas e Informática*, 7(1):57–64, 2014.

- [10] Jorge Enrique Rodríguez, Ángela Paola Herrera Domínguez, and Martha Liliana Rojas Olaya. Sistema de bloqueo automático para páginas web que incitan a la violencia a través de un algoritmo híbrido de aprendizaje computacional. *Revista Vínculos*, 10(2), 2013.
- [11] Paweł Pławiak and Wojciech Maziarz. Classification of tea specimens using novel hybrid artificial intelligence methods. *Sensors and Actuators B: Chemical*, 192:117–125, 2014.
- [12] Qiu-yu Zhang, Hui-yi Wei, Mo-yi Zhang, Zhi-gang Xu, Hong-xiang Duan, and Lu Lv. Research on hand gesture recognition based on inner-distance contour point distribution features and histogram matching. *Journal of Computers*, 9(10):2455–2460, 2014.
- [13] Félix Brezo, José Gaviria de la Puerta, Xabier Ugarte-Pedrero, Igor Santos, Pablo G Bringas, DeustoTech Computing, and David Barroso. A supervised classification approach for detecting packets originated in a http-based botnet. *CLEI ELECTRONIC JOURNAL*, 16(03), 2013.
- [14] Antonio J Tallón-Ballesteros and César Hervás-Martínez. A two-stage algorithm in evolutionary product unit neural networks for classification. *Expert Systems with Applications*, 38(1):743–754, 2011.

## ANEXOS

| Caso USR-001              |  |
|---------------------------|--|
| Nombre de caso de uso     | Ingreso al aplicativo móvil.   |
| Actores, participantes    | Estudiante.  |
| Flujo de Eventos          | 1.- Se da clic en el icono de la aplicación.<br>2.- En la nueva pantalla aparece un pequeño formulario de ingreso, introducir el usuario y la contraseña.<br>3.- Se da clic en el botón "Ingresar".                                |
| Pre Condiciones           | El estudiante tiene instalado la aplicación IDM en su dispositivo móvil.   |
| Post Condiciones          | <b>Ingreso de usuario exitoso</b><br>el sistema mostrará la pantalla principal de la aplicación.<br><b>Ingreso no exitoso</b><br>la aplicación presenta un mensaje donde indica el motivo por el cual no pudo ingresar al sistema. |
| Requerimientos de calidad | Se validará que los campos a llenar no estén vacíos.   |

| Nombre de Escenario    |   |
|------------------------|---|
| Actores, Participantes | Usuario   |
| Flujo de eventos       | 1.- El estudiante ingresa correctamente sus datos.<br>2.- IDM valida que todos los datos sean correctos para ingresar al sistema.<br>3.- Se muestra la pantalla principal del aplicativo móvil. |

| Nombre de Escenario    |   |
|------------------------|---|
| Actores, Participantes | Estudiante  |
| Flujo de eventos       | 1.- El estudiante omite alguna información correspondiente al mismo y proceda dar clic en el botón Ingresar<br>2.- IDM encuentra fallas en el ingreso de datos.<br>3.- Se muestra un mensaje indicando:<br>"Ingrese su usuario" ó "Ingrese su contraseña"<br>dependiendo del campo vacío. |

Tabla 5.1: Caso USR-001

| Caso USR-002              |  |
|---------------------------|--|
| Nombre de caso de uso     | Registrar Estudiante   |
| Actores, participantes    | Estudiante.  |
| Flujo de Eventos          | 1.- Se da clic en la pestaña de “Registrarme”.<br>2.- Aparece una nueva pantalla con el formulario de registro para llenar los siguientes datos: Usuario, Contraseña, Confirmación de contraseña y Correo.<br>3.- Se da clic en el botón “Enviar”.<br>4.- La información se almacena en la base de datos del sistema                 |
| Pre Condiciones           | El estudiante tiene instalada la aplicación IDM en su dispositivo móvil.   |
| Post Condiciones          | <b>Registro almacenado</b><br>El sistema mostrará un mensaje diciendo “Usuario registrado exitosamente”.<br><b>Registro no almacenado</b><br>Se mostrará un mensaje diciendo “El correo electrónico ya se encuentra registrado”.<br>Se mostrará un mensaje por cada campo vacío por ejemplo: usuario, diciendo “Ingrese su usuario”. |
| Requerimientos de calidad | Cualquiera que sea el escenario, la aplicación mostrará un mensaje indicando el resultado de la solicitud.   |

| Nombre de Escenario    | Registro exitoso del estudiante.   |
|------------------------|--|
| Actores, Participantes | Usuario  |
| Flujo de eventos       | 1.- El estudiante ingresa correctamente los datos.<br>2.- IDM valida que todos los datos sean correctos para almacenar la información en la base de datos.<br>3.- Se muestra en un mensaje que el registro fue exitoso |

| Nombre de Escenario    | Registro fallido: Existen campos en blanco  |
|------------------------|---|
| Actores, Participantes | Estudiante  |
| Flujo de eventos       | 1.- El estudiante no omite alguna información correspondiente al mismo y procede dar clic en el botón "Ingresar".<br>2.- IDM encuentra fallos en el ingreso de datos.<br>3.- Se muestra un mensaje indicando: “Ingrese su usuario” ó “Ingrese su contraseña” dependiendo del campo vacío. |

Tabla 5.2: Caso USR-002



| Caso USR-003              |  |
|---------------------------|--|
| Nombre de caso de uso     | Registrar nueva evaluación   |
| Actores, participantes    | Estudiante.  |
| Flujo de Eventos          | <p>1.- Se da clic en el botón “Nueva Evaluación”.</p> <p>2.- En la nueva pantalla se visualiza una imagen correspondiente a una mariposa y tres opciones de géneros para identificar al cual pertenece.</p> <p>3.- Se escoge una opción de género que el usuario cree ser el ítem correcto.</p> <p>5.- Se da clic sobre el botón “Aceptar”.</p> <p>6.- La información se guardará en la base de datos del sistema como nuevo registro de evaluación realizada.</p> |
| Pre Condiciones           | <p>El estudiante se ha autenticado en IDM.</p> <p>El estudiante ha seleccionado la opción "Nueva evaluación".</p>  |
| Post Condiciones          | <p><b>Nueva evaluación guardada</b><br/>La aplicación mostrará un mensaje “Evaluación guardada exitosamente”</p> <p><b>Nueva evaluación no guardada</b><br/>La aplicación presenta un mensaje donde se indica el motivo por el cual no se pudo almacenar la nueva evaluación.</p>  |
| Requerimientos de calidad | Se validará que haya escogido una opción de las que arroja el sistema.   |

| Nombre de Escenario    | Ingreso Exitoso de una nueva evaluación.   |
|------------------------|--|
| Actores, Participantes | Estudiante   |
| Flujo de eventos       | <p>1.- El estudiante escoge una opción.</p> <p>2.- IDM valida la elección de un sólo tipo de género de mariposa y almacena la información en la base de datos.</p> <p>3.- Se muestra un mensaje de que su nueva evaluación ha sido guardada correctamente.</p> |

| Nombre de Escenario    | Ingreso fallido: Opción no seleccionada   |
|------------------------|---|
| Actores, Participantes | Estudiante  |
| Flujo de eventos       | <p>1.- El estudiante no escogió una opción.</p> <p>2.- IDM encuentra fallos en la nueva evaluación.</p> <p>3.- Se muestra un mensaje “Escoja una opción”.</p> |

Tabla 5.3: Caso USR-003

| Caso USR-004              |  |
|---------------------------|--|
| Nombre de caso de uso     | Consulta de Evaluación   |
| Actores, participantes    | Estudiante.  |
| Flujo de Eventos          | 1.- Se da clic en la pestaña "Evaluaciones"<br>2.- Se muestra un listado de las evaluaciones realizadas previamente.<br>3.- Seleccione una evaluación y aparece una nueva pantalla que contiene la información de la misma: Imagen de la especie (mariposa), tres opciones de géneros con un ítem seleccionado anteriormente por el usuario. |
| Pre Condiciones           | El estudiante se ha autenticado en IDM.<br>El usuario debe de seleccionar una evaluación del listado.  |
| Post Condiciones          | <b>Consulta exitosa</b><br>Se muestra la lista de evaluaciones que corresponde al usuario autenticado.<br><b>Consulta no exitosa por falta de datos</b><br>El sistema presentará la lista de evaluaciones vacía.   |
| Requerimientos de calidad | El sistema debe validar que el usuario contenga evaluaciones registrados en la base de datos.  |

| Nombre de Escenario    | Consulta Exitosa de evaluación   |
|------------------------|--|
| Actores, Participantes | Estudiante   |
| Flujo de eventos       | 1.- El estudiante presiona la opción "Evaluaciones".<br>2.- IDM valida qué evaluaciones ha realizado el estudiante<br>3.- Se muestran las evaluaciones realizadas y en ellas la retroalimentación de cada una. |

| Nombre de Escenario    | Consulta vacía: No existen evaluaciones  |
|------------------------|--|
| Actores, Participantes | Estudiante   |
| Flujo de eventos       | 1.- El estudiante presiona la opción "Evaluaciones".<br>2.- IDM valida que evaluaciones ha realizado el estudiante<br>3.- No se encuentran registros de evaluaciones realizadas por el estudiante. |

Tabla 5.4: Caso USR-004

| <b>Caso ADM-001</b>       |  |
|---------------------------|--|
| Nombre de caso de uso     | Autenticación  |
| Actores, participantes    | Experto.   |
| Flujo de Eventos          | <ol style="list-style-type: none"> <li>1.- El usuario ingresa su identificador de usuario y contraseña.</li> <li>2.- Se hace clic en el botón "Ingresar".</li> <li>3.- Administrador IDM valida que todos los campos estén llenos y validados.</li> <li>4.- Si los campos no están correctamente validados se mostrarán errores y no se permitirá el ingreso.</li> <li>5.- Si el usuario y contraseña es correcto se ingresa al portal del administrador IDM.</li> </ol> |
| Pre Condiciones           | El usuario se encuentra registrado en el sistema.  |
| Post Condiciones          | <p>En caso de haber campos vacíos, el sistema mostrará un mensaje diciendo "Credenciales incorrectas".</p> <p>El sistema no se pudo enlazar con el servidor.</p>   |
| Requerimientos de calidad | Cualquiera sea el escenario, el sistema mostrará un mensaje indicando el resultado de la solicitud.  |

| <b>Nombre de Escenario</b>         |   |
|------------------------------------|---|
| <b>Ingreso exitoso al sistema.</b> |   |
| Actores, Participantes             | Experto   |
| Flujo de eventos                   | <ol style="list-style-type: none"> <li>1.- Ingresar su usuario y contraseña.</li> <li>2.- Administrador IDM valida que todos los datos sean correctos.</li> <li>3.- Se muestra la ventana de Inicio con todas las opciones permitidas para el usuario.</li> </ol> |

| <b>Nombre de Escenario</b>                     |  |
|--|--|
| <b>Ingreso fallido.- Existen campos vacíos</b> |  |
| Actores, Participantes                         | Experto  |
| Flujo de eventos                               | <ol style="list-style-type: none"> <li>1.- El estudiante presiona la opción "Evaluaciones".</li> <li>2.- IDM valida las evaluaciones que han sido realizadas por el estudiante</li> <li>3.- No se encuentran evaluaciones realizadas por el estudiante.</li> </ol> |

Tabla 5.5: Caso USR-005