



# UNIVERSIDAD DEL PAPALOAPAN

Campus Loma Bonita

---

LICENCIATURA EN MATEMÁTICAS APLICADAS

## ANÁLISIS DE IMÁGENES POR BORDES APLICADO A LA COMPARACIÓN MORFOLÓGICA DE TORTUGAS MARINAS

### TESIS

QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADA EN MATEMÁTICAS APLICADAS

#### PRESENTA:

ERIKA MAULEÓN DOMÍNGUEZ

#### ASESOR DE TESIS:

DR. EDUARDO SÁNCHEZ SOTO

#### CO-ASESOR DE TESIS:

DR. SERGIO IVVAN VALDÉS PEÑA



**UNIVERSIDAD DEL PAPA LOAPAN**  
**CAMPUS LOMA BONITA**

LOMA BONITA, OAXACA, 2016

**LICENCIATURA EN MATEMÁTICAS APLICADAS**

LA PRESENTE TESIS TITULADA

**“ANÁLISIS DE IMÁGENES POR BORDE APLICADO A LA COMPARACIÓN MORFOLÓGICA DE TORTUGAS MARINAS”**

PRESENTADA POR LA SUSTENTANTE DE LICENCIATURA: **C. ERIKA MAULEÓN DOMÍNGUEZ** BAJO LA ASESORÍA DEL **DR. SERGIO IVVAN VALDEZ PEÑA** Y EL **DR. EDUARDO SÁNCHEZ SOTO**, HA SIDO REVISADA Y ACEPTADA POR EL COMITÉ EXAMINADOR PARA SER DEFENDIDA EN EL EXAMEN PROFESIONAL Y OBTENER EL TÍTULO DE **LICENCIADA EN MATEMÁTICAS APLICADAS**.

**DR. EDUARDO SÁNCHEZ SOTO**

ASESOR

**DR. SERGIO IVVAN VALDEZ PEÑA**

CO-ASESOR

**M.C. EDUARDO ORTIZ HERNÁNDEZ**

PRESIDENTE

**DR. JUAN MANUEL PÉREZ ABARCA**

SECRETARIO

**DR. SERGIO IVVAN VALDEZ PEÑA**

VOCAL

*Dedicado a  
mis padres*

# Agradecimientos

*Gracias Edith Domínguez Reyes por todo tu esfuerzo y sacrificio, por tu infinito amor incondicional. Por apoyarme siempre, a luchar por mis sueños. Por ser el motivo principal de mis deseos de superación, por eso y mucho más... mil gracias mamá. Te amo y siempre estaré en deuda contigo madre. Gracias Manuel Mauleón Álvarez, por haberme apoyado y amado, por ser ahora mi ángel guardián. Te extraño y nunca dejaré de amarte padre.*

*Gracias Dios por haberme permitido cumplir uno de mis sueños al lado de mi familia. Gracias a mis abuelitas, tíos y tías que me apoyaron en esta etapa de mi vida. A mi sobrinita Nazli Nahomi a quien adoro con todo mi corazón. A mi hermano Manuel a quien amo, a pesar de todo.*

*A mi director de tesis Dr. Eduardo Sánchez Soto, por su tiempo y paciencia. Por su conocimiento, el cual fue de gran apoyo en la realización de esta tesis. Al Dr. Juan Manuel Pérez Abarca, porque sus asesorías me ayudaron a visualizar de forma diferente mis dudas e inquietudes.*

*A mis revisores el M.C. Eduardo Ortiz Hernández, el Dr. Juan Manuel Pérez Abarca, el Dr. Sergio Ivvan Valdés Peña y el Dr. Eduardo Sánchez Soto por sus correcciones y opiniones que sin duda fueron de gran ayuda.*

*A tres hermosas criaturas por hacerme sonreír. Gracias a todos los que de alguna manera u otra contribuyeron en la realización de unas de mis metas.*

# Resumen

En este trabajo se propone una distancia basada en los códigos de cadena que nos permite clasificar imágenes. Dichas cadenas se obtuvieron de los bordes de las figuras que se desean clasificar después de haber sido binarizadas, por un umbral de decisión tomado del histograma de cada imagen. Antes de efectuar la distancia propuesta se alinean las cadenas por los algoritmos de alineamiento temporal dinámico o Needleman-Wunsch, debido a que dichos algoritmos garantizan un alineamiento óptimo, lo cual es necesario para obtener una adecuada clasificación de imágenes.

Se efectuó la distancia propuesta en la clasificación de imágenes de tortugas marinas. En específico, se analizaron las imágenes de las tortugas carey y verde. Aunque existen varias formas de identificar a cada especie, en esta tesis se clasificaron utilizando dos de las escamas que las diferencian entre sí. Cabe mencionar que la distancia propuesta es de primordial importancia para lograr una clasificación apropiada en las imágenes de tortugas marinas.

**Palabras clave:** *Análisis de imágenes, códigos de cadena y alineamiento de secuencias.*

# Abstract

In this study, a distance based on chain codes that allows us to classify images is proposed. The said chains were obtained on the borders of the figures which are to be classified after having been binarized, for a threshold decision made by the histogram of every image. Before effecting the proposed distance an alignment of the chains was necessary for the algorithms of Dynamic Time Warping or Needleman-Wunsch, this is to say that the algorithms guarantee an optimal alignment, which is necessary to obtain an adequate image of classification.

The distance was effected in the classification of images of sea turtles, to be specific, the images of the hawksbill and green turtle were analyzed. Although various forms of identifying each species exist, in this study they were classified by using two of the scales that differ from one another. It's important to mention that the distance proposed is of paramount importance for obtaining an appropriate classification of the images of sea turtles.

**Keywords:** *Image analysis, chain codes and sequence alignment.*

# Índice general

<b>Agradecimientos</b>	<b>II</b>
<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>IV</b>
<b>Lista de figuras</b>	<b>VI</b>
<b>Lista de tablas</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.2. Objetivos . . . . .	4
1.2.1. Objetivo general . . . . .	4
1.2.2. Objetivos específicos . . . . .	4
1.3. Hipótesis . . . . .	5
1.4. Justificación . . . . .	5
1.5. Alcances y limitaciones . . . . .	6
<b>2. Marco de referencia</b>	<b>7</b>
2.1. Antecedentes . . . . .	7
2.2. Tortugas Marinas . . . . .	11

<i>ÍNDICE GENERAL</i>	VI
2.2.1. Características para la Identificación de las especies . . . . .	12
2.2.2. Tortugas Carey . . . . .	13
2.2.3. Tortuga Verde . . . . .	14
2.3. Morfología . . . . .	15
2.4. Análisis de imágenes . . . . .	15
2.4.1. Procesamiento de imágenes . . . . .	16
2.4.2. Reconocimiento de patrones . . . . .	16
2.4.2.1. Estructura en un sistema de reconocimiento de patrones. . . . .	17
2.5. Programación Dinámica . . . . .	18
<b>3. Código de Cadena</b>	<b>21</b>
3.1. Código de Cadena por Vértice (VCC) . . . . .	22
<b>4. Alineamiento de Secuencias</b>	<b>24</b>
4.1. Alineamiento global y local . . . . .	24
4.2. Algoritmos de alineamiento . . . . .	25
4.2.1. Método de la matriz de puntos . . . . .	25
4.2.2. Método de programación dinámica . . . . .	26
4.2.2.1. Traceback . . . . .	26
4.2.2.2. Algoritmo de Needleman Wunsch . . . . .	28
4.2.2.3. Algoritmo de Alineamiento Temporal Dinámico . . . . .	29
4.2.3. Método heurístico de palabra corta . . . . .	31
<b>5. Distancia propuesta</b>	<b>32</b>
<b>6. Protocolo experimental</b>	<b>37</b>
6.1. Preprocesamiento de las imágenes . . . . .	37
6.1.1. Prueba 1 de la binarización . . . . .	39
6.1.2. Prueba 2 de la binarización . . . . .	39

<i>ÍNDICE GENERAL</i>	VII
6.2. Distancias de Cadenas alineadas de polígonos . . . . .	40
6.3. Comparación entre la tortuga Verde y Carey . . . . .	43
6.3.1. Comparación de las escamas 1 de las tortugas carey y verde . . . . .	44
6.3.2. Comparación de las escamas 2 de las tortugas carey y verde . . . . .	50
<b>7. Conclusiones</b>	<b>57</b>
<b>Bibliografía</b>	<b>58</b>

# Índice de figuras

2.1. Aleta con marca plástica . . . . .	7
2.2. Aleta con grapa metálica . . . . .	7
2.3. Tortuga verde con una marca viva. En el plastrón, la marca se ve como un punto oscuro (o una raya) y en el caparazón, como un punto claro (o una raya). . . . .	8
2.4. Insertando un PIT en la aleta delantera de una tortuga marina, mostrando ambos puntos de inserción (a) ventral y (b) dorsal. . . . .	8
2.5. A) El rectángulo amarillo muestra la marca rosada en la Tortuga Laúd. B) Arriba: visualización de los puntos claves. Abajo: vectores comparados mediante el algoritmo (modificado de Zeeuw) . . . . .	10
2.6. Codificación de las escamas faciales de la tortuga carey . . . . .	10
2.7. Comparación potencial de una hembra de <i>C. mydas</i> propuesta por el software Wild-ID.	11
2.8. Escamas de la cabeza . . . . .	12
2.9. Escudos del caparazón y del plastrón . . . . .	12
2.10. Tortuga Carey . . . . .	13
2.11. Tortuga Verde . . . . .	14
2.12. Estructura en un sistema de reconocimiento de patrones . . . . .	17
3.1. Código de cadena por dirección . . . . .	21
3.2. Código de cadena por vértice [12]. . . . .	23

6.1. Diagrama de flujo de la condición que se utiliza para binarizar. (Nota: Todos los valores son adquiridos del histograma) . . . . .	38
6.2. Binarización con umbral 1 . . . . .	39
6.3. Binarización con umbral 2 . . . . .	39
6.4. Binarización con umbral 1 . . . . .	40
6.5. Binarización con umbral 2 . . . . .	40
6.6. Representación gráfica de distancias resultantes de las cadenas alineadas. . . . .	42
6.7. Escamas 1 y 2 de la tortuga carey . . . . .	43
6.8. Escamas 1 y 2 de la tortuga verde . . . . .	43
6.9. Ejemplo de escama 1 de la tortuga carey . . . . .	44
6.10. Ejemplo de escama 1 de la tortuga verde . . . . .	44
6.11. Binarización de la figura 6.9 . . . . .	44
6.12. Binarización de la figura 6.10 . . . . .	44
6.13. Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.9. . . . .	48
6.14. Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.9. . . . .	48
6.15. Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.10. . . . .	49
6.16. Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.10. . . . .	49
6.17. Escama 2 de la tortuga carey . . . . .	50
6.18. Escama 2 de la tortuga verde . . . . .	50
6.19. Binarización de la figura 6.17 . . . . .	50
6.20. Binarización de la figura 6.18 . . . . .	50
6.21. Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.17. . . . .	54

*ÍNDICE DE FIGURAS*

X

6.22. Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.17. . . . .	55
6.23. Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.18. . . . .	55
6.24. Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.18. . . . .	56

# Índice de tablas

4.1. Ejemplo de alineamiento global y local. . . . .	25
4.2. Matriz de distancias mínimas de dos secuencias de palabras . . . . .	28
5.1. Ejemplo de alineamiento de secuencias con el algoritmo de Needleman-Wunsch, y sus respectivas distancias . . . . .	35
6.1. Valores que utiliza el programa para binarizar y en la condición de binarización . . . . .	39
6.2. Valores que utiliza el programa para binarizar y en la condición de binarización . . . . .	40
6.3. Distancias resultantes de las cadenas alineadas (de los diferentes polígonos) por los algoritmos ATD y Needleman-Wunsch. . . . .	42
6.4. Medias de las distancias obtenidas con el análisis de las escamas 1 de las tortugas carey y verde. . .	47
6.5. Medias de las distancias obtenidas con el análisis de las escamas 2 de las tortugas carey y verde. . .	54

# Capítulo 1

## Introducción

Desde hace más de 100 millones de años las tortugas marinas habitan en nuestro planeta. En los últimos años a disminuido drásticamente la población en la mayoría de las especies de tortugas marinas, a tal grado que algunas especies de tortugas se encuentran en peligro de extinción, y peor aún según la Lista Roja de la Unión Interamericana para la Conservación de la Naturaleza (UICN) [27], existen algunas especies de tortugas marinas que se encuentran en estado crítico de extinción. Algunos factores son:

- El consumo de la carne de la tortuga.
- El consumo de sus huevos, por parte de los animales y humanos.
- La comercialización de su caparazón.
- Cambios climáticos.
- La contaminación por químicos, desechos sólidos y líquidos.
- La destrucción de áreas de alimentación, anidación, reposo y hábitat.

Hoy en día la ley prohíbe su consumo; además existen áreas naturales protegidas con la finalidad de conservar las especies de tortugas marinas.

Existen varias formas de identificar una tortuga marina; por ejemplo, por medio de su cráneo o por la estructura de su caparazón y cabeza; así como otros rasgos característicos de algunas especies que ayudan a una mejor clasificación. De lo anterior, podemos concluir que se pueden identificar y clasificar a las tortugas marinas por medio de imágenes.

Una imagen es una colección de objetos y regiones aunada a la interrelación que existe entre ellos [16]. Esto implica un gran potencial de información que demanda avanzadas técnicas matemáticas, físicas y computacionales. Las personas se encuentran rodeadas de imágenes que observan por medio del sentido de la vista; obtenemos información en forma de imágenes de todo tipo, en todos colores y en blanco y negro. Las personas son capaces de analizar e interpretar de manera extremadamente rápida las imágenes que recibe. Varias decisiones que una persona toma son inmediatas, sobre la marcha, y están fundamentadas en la información que extrae de las imágenes que recoge del mundo que le rodea.

Las funciones cerebrales de extracción de objetos de una escena no están cabalmente comprendidas y por lo mismo no han podido ser reproducidas en sistemas automáticos de análisis digital. El objetivo no es copiar tales funciones cerebrales, sino entenderlas y emularlas en procesos automáticos eficientes. En ciertas tareas muy específicas, el análisis automático es equiparable o incluso supera la función cerebral correspondiente, tal es el caso de la detección de bordes en imágenes complejas [16].

Los problemas comunes (entre otros) a los que se enfrentan los diferentes investigadores son la caracterización de sus datos, el realce de sus patrones y el reconocimiento de los mismos.

La finalidad en el análisis de imágenes es la cuantificación de las propiedades de los objetos presentes en éstas, para lo cual es necesario entender los elementos que las componen, su origen y

su naturaleza. Los métodos de análisis están estrechamente relacionados con estos aspectos. La toma de decisiones en áreas como la medicina, la astronomía, geología, biología, se ayudan del resultado de este análisis.

La utilidad del procesamiento de imágenes es muy amplia y abarca muchos campos. Un ejemplo son las imágenes obtenidas con el fin de identificar y clasificar los diferentes tipos de especies de tortugas marinas que habitan en el planeta. En este trabajo nos proponemos automatizar una parte del procedimiento con la ayuda de una computadora, los códigos de cadena [12] y la propuesta de una distancia que nos ayuda a la clasificación.

En el capítulo 2 se proporciona información referente a las especies de tortugas que se van a clasificar, al análisis de imágenes y a la programación dinámica. En los capítulos 3 y 4, se facilita una breve explicación de códigos de cadena y alineamiento de secuencias, respectivamente.

En el capítulo 5, se plantea la propuesta de una distancia que nos permite clasificar las cadenas obtenidas de las formas en las imágenes. En el capítulo 6, se muestran algunos resultados y en el capítulo 7 se exponen las conclusiones de esta tesis.

## **1.1. Planteamiento del problema**

El problema que nos planteamos es el de proponer una distancia que nos permita identificar imágenes, utilizando el código de cadena correspondiente al borde de las figuras que se desean clasificar. En este caso, utilizaremos imágenes de tortugas marinas y las cadenas que obtendremos serán de las escamas que diferencian a las especies y hasta cierto grado a los individuos.

Dicho proceso suele ser complejo, esto debido a la escasa recopilación de imágenes que son

necesarias para un modelo robusto en el reconocimiento de las especies. Algunos de los problemas encontrados tienen que ver con que algunas imágenes son más oscuras, otras más claras y otras más suelen tener en ciertas regiones más luminosidad (esto debido a los rayos del sol o al flash de la cámara); es decir, no todas las imágenes cuentan con la misma calidad, lo cual dificulta la identificación de las especies de tortugas marinas. El pre-proceso para subsanar estos problemas se llevó a cabo de manera semiautomática y se puso especial énfasis en el modelo con los códigos de cadena.

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Proponer una distancia basada en los códigos de cadenas que sea eficiente para la clasificación de formas en imágenes.

### **1.2.2. Objetivos específicos**

Los objetivos específicos que nos permitirán lograr el objetivo general se describen a continuación:

- Aprender todos los conceptos relacionados con el análisis de imágenes.
- Calcular el código de cadena correspondiente de las imágenes segmentadas, elaborando un algoritmo en Matlab..
- Proponer una distancia que clasifique de manera exitosa las imágenes.
- Realizar un programa que nos permita alinear secuencias.
- Estudiar la morfología de las tortugas marinas; para poder diferenciar las tortugas carey de otras especies de tortugas marinas.

- Identificar los patrones y/o bordes importantes de las imágenes de tortugas marinas.
- Programar el algoritmo para resaltar los bordes de una imagen (segmentación).

### 1.3. Hipótesis

Se planteará una distancia que nos permita clasificar imágenes. Primero se procesaran las imágenes para obtener un realce en los bordes de las figuras. Suponiendo que los códigos de cadena son eficientes, se obtendrán los códigos de cadena de las características importantes de las figuras, los cuales nos ayudan a diferenciarlas de otras figuras (objetos, personas, animales, entre otras). Para utilizar la distancia que obtendremos, debemos alinear dichas cadenas. Así esperamos que los resultados sean más eficientes; todo lo anterior se realizará en algún lenguaje de programación.

### 1.4. Justificación

La sencillez, invarianza a rotaciones y translaciones, reducción en la información y conservación de las características importantes en las imágenes generará un algoritmo rápido, robusto y eficiente para su implementación en dispositivos móviles. Los robots de cámara son de gran utilidad para obtener información mediante las fotografías; una de sus aplicaciones puede ser adquirir imágenes de tortugas.

En los últimos años ha disminuido enormemente las poblaciones de tortugas marinas, por ello los especialistas desarrollaron la técnica de marcaje-recaptura la cual sirve para tener una mejor identificación de las tortugas marinas que habitan en el planeta. Sin embargo, dicha técnica tiene algunas desventajas, como el alto costo y la pérdida de las marcas metálicas. Debido a lo anterior

se ha buscado nuevas técnicas de identificación. El análisis de imágenes es un método alternativo que sin duda será de gran ayuda para identificar y clasificar las especies de tortugas marinas.

## 1.5. Alcances y limitaciones

**Los alcances de este trabajo son:**

- Proponer una distancia que nos permita comparar objetos, utilizando códigos de cadenas y alineamiento de secuencias.
- Binarizar de forma automática las imágenes y obtener la cadena correspondiente.
- Obtener un algoritmo que alinee dos pares de cadenas; y posteriormente calculé su distancia.
- Clasificar las escamas que diferencian a las tortugas Carey y verde, usando la distancia propuesta.

**Las limitaciones de este trabajo son:**

- No se detectaron de modo automático las zonas de interés; es decir, se recortaron las escamas que nos ayudan a clasificar a las dos especies de tortugas marinas.
- La cantidad de imágenes de tortugas Carey y verdes es insuficiente, debido a esto la distancia media que se calculó sólo sirve para darnos una idea del rango en el cual se encuentran las distancias medias de dos especies de tortugas marinas; es decir, sólo es una aproximación de la distancia media real que debe existir entre dos imágenes de la misma especie y dos de diferentes especies.

## Capítulo 2

# Marco de referencia

### 2.1. Antecedentes

Con el fin de la conservación de las especies, los especialistas han implementado el sistema de marcaje-recaptura, el cual permite identificar y tener una mejor idea de la evolución de las tortugas a través del tiempo. El marcado de tortugas marinas puede efectuarse por medio de marcas externas o internas.

Algunas marcas externas son:

- Las grapas metálicas o plásticas, inscritas con números y letras, las cuales se aplican en los bordes posteriores de las aletas, por medio de una pinza especial.



Figura 2.1: Aleta con marca plástica



Figura 2.2: Aleta con grapa metálica

- Marcas de pigmentación contrastante por medio del intercambio quirúrgico (o autotransplante) de pequeños fragmentos de tejido entre el caparazón y el plastrón. Estas marcas son llamadas a veces “marcas vivientes”.



Figura 2.3: Tortuga verde con una marca viva. En el plastrón, la marca se ve como un punto oscuro (o una raya) y en el caparazón, como un punto claro (o una raya).

Algunas marcas internas son:

- La inserción en el cuerpo de una marca de alambre, dichas marcas pueden magnetizarse pasando un imán sobre ellas antes o después de ser implantadas, si se usa un magnetómetro para su detección.
- Los transmisores pasivos integrados (PIT) son pequeños microprocesadores que puede ser detectado con un instrumento electrónico.



Figura 2.4: Insertando un PIT en la aleta delantera de una tortuga marina, mostrando ambos puntos de inserción (a) ventral y (b) dorsal.

Algunas de las desventajas de usar marcas tradicionales son: el alto costo, la pérdida de las marcas, el malestar o dolor que causa la aplicación de las marcas externas o internas, el desgaste de los números grabados en las marcas plásticas hasta el punto de hacerse imposible leerlas [11], según algunos investigadores se podría estar perjudicando la habilidad de la tortuga para orientarse (si se utilizan las marcas magnetizadas) y la imposibilidad de reconocimiento de una tortuga marcada por alguien que no cuente con un lector (en el caso de las marcas PIT) [8].

Hoy en día existe otra técnica para identificar a las especies, por ejemplo la foto-identificación, la cual ha resultado ser muy eficaz. Debido a que permite reconocer a las especies a partir de características físicas por medio de imágenes, la foto-identificación emplea imágenes para observar dichas características físicas destacadas del animal que permiten el reconocimiento individual. En resumen la foto-identificación es la técnica que permite identificar a los individuos por sus marcas naturales a través de la fotografía.

Uno de los trabajos en el cual se ha implementado la técnica foto-identificación es el presentado por Zeeuw y colaboradores (2010), donde se identificaron hembras de tortuga laúd (*Dermochelys coriacea*) a partir de la comparación de imágenes con la mancha rosada, que se encuentra en la parte superior de la cabeza. En dicho estudio se elaboró un software que permitió identificar a las hembras, aunque las imágenes presentaban variaciones de iluminación, resolución y ángulo. El sistema de cómputo propuesto se basó en el uso del algoritmo SIFT [28].

El algoritmo SIFT realiza comparaciones entre pares de imágenes mediante la selección de puntos clave independientes al escalado y rotación. Estos puntos clave son caracterizados por medio de descriptores vectoriales, a partir de los cuales se establece la similitud entre los pares de imágenes.

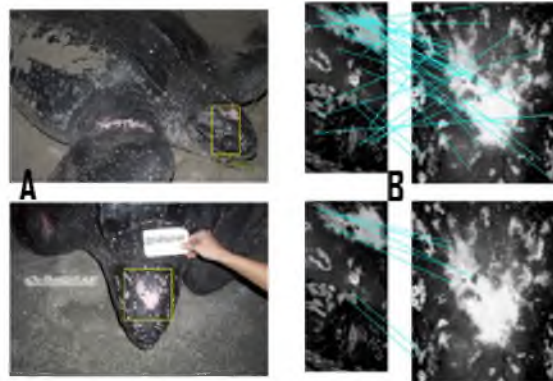


Figura 2.5: A) El rectángulo amarillo muestra la marca rosada en la Tortuga Laúd. B) Arriba: visualización de los puntos claves. Abajo: vectores comparados mediante el algoritmo (modificado de Zeeuw)

Otro estudio de interés que ha utilizado la foto-identificación es el realizado por Jean y colaboradores en individuos de tortuga Carey y verde desde Reunión, Mayotte y Mahe en el Océano Índico occidental; usando la codificación de los perfiles faciales de acuerdo con la posición y la forma de las escamas utilizando fotografías y un asistente de cómputo. Cada perfil facial de la tortuga marina se transforma mediante inspección visual en un código. Este código consta de cuatro dígitos; el primer dígito representa al número de escamas postoculares, continuado con una serie de código formada por tres números, el primero pertenece al número de fila, el segundo corresponde a la posición del escudo en esa fila y tercero corresponde al número de lados del escudo [14].



Figura 2.6: Codificación de las escamas faciales de la tortuga Carey

Labastida Estrada Elizabeth usó la foto-identificación en poblaciones de tortugas Carey y blanca, utilizando como marca natural el patrón de la forma, arreglo y coloración de las escamas

faciales de cada individuo [11]. Las fotografías se utilizaron en formato digital y la comparación entre imágenes se realizó mediante el software Wild-ID, con el cuál se generan patrones de comparación a partir del algoritmo SIFT (Scale invariant Feature Transform).



Figura 2.7: Comparación potencial de una hembra de *C. mydas* propuesta por el software Wild-ID.

El software de la foto-identificación como Wild-ID se puede utilizar para facilitar las identificaciones individuales. El programa Wild-ID fue desarrollado por ingenieros de software en el colegio Dartmouth, que puede rápidamente ordenar a través de muchas de nuestras fotografías para decirnos cuál de los patrones hemos visto anteriormente.

## 2.2. Tortugas Marinas

Las tortugas marinas se agrupan en dos familias. La familia Cheloniidae incluye las que tienen el caparazón cubierto con escamas. La familia Dermochelyidae incluye sólo una especie moderna de tortugas marinas, la tortuga baula (*Dermochelys coriacea*), cuyo caparazón es coriácea.

Cada tortuga marina tiene un nombre científico y muchos nombres comunes. El nombre científico identifica el género y la especie; y los nombres comunes muchas veces describen características de la tortuga.

### 2.2.1. Características para la Identificación de las especies

La identificación de las tortugas marinas, usando características externas, se basa en las escamas de la cabeza, la forma de las mandíbulas, el número de uñas en las aletas, y el número y arreglo de las placas o escudos en el caparazón [15].

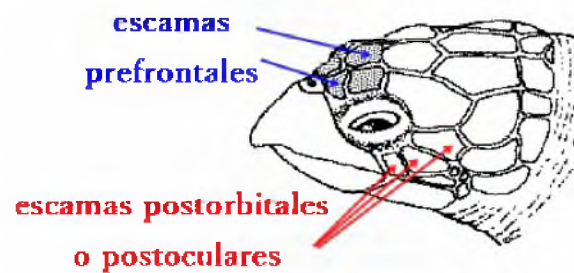


Figura 2.8: Escamas de la cabeza

La concha inferior se denomina plastrón. El plastrón también tiene características distintivas en los escudos, pero estos se usan más como localizadores de estructuras internas y no tanto para la identificación de especies [15].

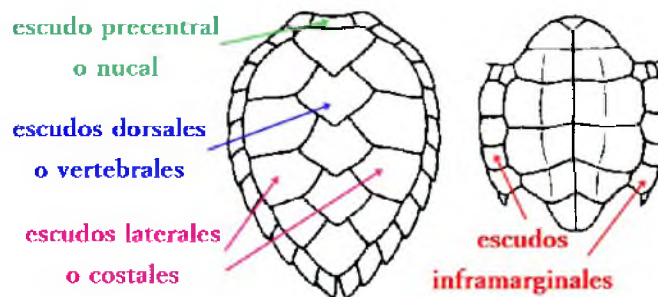


Figura 2.9: Escudos del caparazón y del plastrón

### 2.2.2. Tortugas Carey



Figura 2.10: Tortuga Carey

Nombre científico: *Eretmochelys imbricata*.

Según la Lista Roja de la Unión Interamericana para la Conservación de la Naturaleza (UICN), la tortuga carey se encuentra en peligro crítico de extinción.

La tortuga carey es la única tortuga marina que sus escamas del caparazón traslapan (imbricados) con las que están detrás de ellas; es decir, su caparazón está formado por placas sobrepuestas. El caparazón presenta trece escudos agrupados en cinco dorsales y cuatro pares laterales, el escudo nucal no toca el primer escudo lateral. El plastrón presenta 4 escudos inframarginales sin poros. Y las aletas anteriores (delanteras) presentan dos uñas.

Su coloración cuando recién nacida es marrón caoba oscuro tanto en el caparazón como en el plastrón; a medida que la tortuga va creciendo, la cabeza se alarga y el caparazón va desarrollando un patrón distintivo de rayos en amarillo, negro, canela y marrón en cada escudo, este color persiste en el adulto. La parte ventral va del color crema al amarillo claro. De forma general se podría decir que el caparazón es de color café oscuro o café amarillo ámbar.

La cabeza cuenta con dos pares de escamas prefrontales y tres pares de escamas postorbitales, su cráneo es largo y angosto en todas las edades, excepto en los recién nacidos, su longitud es aproximadamente igual a dos veces su anchura. Mandíbulas formadas como un pico, dicho pico es córneo, angosto y filoso; similar al de un halcón. El pico les permite a las tortugas conseguir alimento que se encuentra en las grietas de los arrecifes de coral.

Las crías recién emergidas con la cicatriz umbilical miden alrededor de 6 cm de longitud del caparazón, las cuales tienen hábitos pelágicos hasta alcanzar los 25 cm aproximadamente. Se les considera juveniles de 25-55 cm aproximadamente, subadultos de 55 a 70 cm y los adultos (maduros sexualmente) de 70 cm en adelante. En general, la longitud total en línea recta va de los 76 a los 114 cm. Y su caparazón mide entre 60 y 95 cm.

La tortuga carey presenta dimorfismo sexual evidente en los adultos, con los machos presentando una cola larga y gruesa, extendiéndose mucho más allá del margen posterior del caparazón, también cuentan con uñas largas y fuertes y el plastrón es suave y cóncavo. Las hembras, por el contrario, tienen una cola corta que no se extiende por mucho fuera del margen del caparazón, las uñas son cortas y delgadas mientras que el plastrón es duro y menos cóncavo que el de los machos.

### 2.2.3. Tortuga Verde



Figura 2.11: Tortuga Verde

Nombre científico: *Chelonia mydas*.

Según la Lista Roja de la Unión Interamericana para la Conservación de la Naturaleza (UICN), la tortuga carey se encuentra en peligro de extinción.

Es la única especie herbívora entre las tortugas marinas y se alimenta de pastos marinos y algas. Se le llama Tortuga Verde debido al color de la grasa ubicada bajo su caparazón. El caparazón es liso con escudos o placas evidentes, tiene forma de corazón; y presenta trece escudos agrupados en cinco dorsales y cuatro pares laterales que a veces son irregulares aunque no aserrados, el escudo nual no toca el primer escudo lateral. Las aletas anteriores (delanteras)

presentan una uña.

El color del caparazón cambia con la edad de la tortuga. Es negro en las tortugas recién nacidas, cambiando para marrón con canela en los juveniles, y verde olica o verde-grisáceo en los adultos, a veces con pequeñas manchas amarillas y marrones [15]. La coloración de las tortugas del Pacífico es más oscura.

La cabeza es pequeña en comparación con el tamaño del cuerpo, redonda y con escamas bien definidas, cuenta con un par de escamas prefrontales y cuatro pares de escamas postorbitales. Tiene el borde de la mandíbula inferior aserrado, el pico es corto, y su cráneo mide aproximadamente 15 cm de ancho.

Es la tortuga más grande de la familia Cheloniidae. Los adultos miden hasta 1.5 metros de longitud. Las tortugas negras o verdes del Pacífico son más pequeñas, miden entre 70 y 90 metros.

### **2.3. Morfología**

Es la rama de la biología que se ocupa del estudio y la descripción de las formas externas de un objeto. Dentro de la Biología, la morfología se ocupa del estudio de las formas y estructuras que constituyen a los seres vivos. Su abordaje puede tener un interés específicamente descriptivo o comparativo. La palabra morfología se compone de los vocablos griegos *μορφή* (*morphé*), que significa 'forma', y *λόγος* (*lógos*), 'tratado' [24].

### **2.4. Análisis de imágenes**

El análisis de imágenes consiste en la cuantificación de las propiedades morfológicas y estadísticas de patrones, antecedida por una transformación geométrica a la imagen que los pone

en evidencia. Entonces, el procesamiento de imágenes es aquel conjunto de transformaciones geométricas diseñadas para realzar uno o varios patrones, mientras que el reconocimiento de los mismos sólo se hace a través de los modelos matemáticos que caracterizan sus diferentes aspectos morfológicos y estadísticos.

El reconocimiento de patrones sigue generalmente a la transformación geométrica, de tal forma que si a éste lo expresamos simbólicamente como  $O_{rp}$  y al procesamiento de imágenes como  $O_{pi}$ , tenemos entonces que el análisis de imágenes se expresa como

$$\text{Análisis de imágenes} = O_{rp}\{O_{pi}(g)\}$$

Es decir, el operador  $O_{pi}$  realza los patrones de interés y opera sobre la imagen  $g$  antes que el operador  $O_{rp}$ , el cual cuantifica sus propiedades [16].

### 2.4.1. Procesamiento de imágenes

El procesamiento de imágenes tiene como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar. La imagen puede haber sido generada de muchas maneras, por ejemplo, fotográficamente, o electrónicamente, por medio de monitores de televisión. El procesamiento de las imágenes se puede en general hacer por medio de métodos ópticos, o bien por medio de métodos digitales, en una computadora [6].

### 2.4.2. Reconocimiento de patrones

Una imagen digital contiene cierta riqueza de información sobre la escena observada; el objeto del reconocimiento de patrones consiste en sintetizar y presentar dicha información de una manera que sirva como entrada al subsistema que genera los modelos de la escena y al comportamiento del sistema físico correspondiente [17].

Definiciones de reconocimiento de patrones:

1. Es la búsqueda de estructuras (clases) en la imagen, donde ésta se encuentra formada por todos los datos (píxeles) generados por el sistema sensor [16].
2. Es el resultado de haber aplicado una o varias métricas a las propiedades que caracterizan los patrones o clases de patrones presentes en una imagen; es decir, el reconocimiento de patrones es básicamente un proceso de medida aplicado a la imagen [16]. Las propiedades y las métricas deben ser previamente definidas por medio de modelos matemáticos pertinentes.

Sea  $S$  un conjunto de patrones y  $R$  los números reales,  $\forall s_1, s_2, s_3 \in S$  una métrica es una distancia  $d : S \times S \rightarrow R$  que satisface las siguientes condiciones:

$$d(s_1, s_2) \geq 0, \quad \text{positividad}$$

$$d(s_1, s_2) = 0 \Leftrightarrow s_1 = s_2$$

$$d(s_1, s_2) = d(s_2, s_1), \quad \text{simetría}$$

$$d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3), \quad \text{desigualdad del triángulo}$$

#### 2.4.2.1. Estructura en un sistema de reconocimiento de patrones.

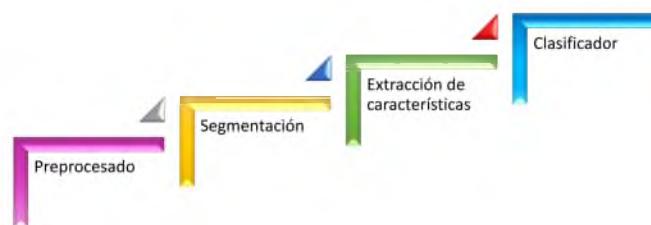


Figura 2.12: Estructura en un sistema de reconocimiento de patrones

**Preprocesado:** Pretende mejorar o realzar las propiedades de la imagen (por ejemplo, mejorar el brillo, realzar o aumentar el contraste, suavizar o eliminar el ruido) para facilitar las siguientes

etapas del reconocimiento de la imagen.

**Segmentación:** Segmentar una imagen digital significa dividirla en zonas disjuntas e individualizadas. Es decir, consiste en diferenciar los diversos objetos y dónde se encuentran del fondo, que puede ser más o menos complejo, de la imagen [1]. En dicho módulo se binarizó de forma automática las imágenes.

**Extracción de Características:** Se encarga de extraer la información que mejor caracterizan a los objetos que serán clasificados, eliminando la información redundante e irrelevante. Su principal propósito es reducir la dimensionalidad del problema de reconocimiento de patrones [20]. En este módulo utilizamos los códigos de cadena y alineamiento de secuencias.

**Clasificador:** El sistema decide a que categoría (clase) pertenece el objeto. La distancia propuesta es el clasificador en esta tesis.

## 2.5. Programación Dinámica

La programación dinámica es un método para la resolución de problemas complejos desglosándolos en subproblemas más simples de manera recursiva [25]. La idea en general de la programación dinámica es resolver en diferentes partes un problema (subproblemas), para después combinar las soluciones de los subproblemas y conseguir una solución global.

Un algoritmo de programación dinámica resuelve cada subproblema una sola vez y después guarda o almacena su respuesta, de este modo se reduce el trabajo de recalcularse la solución la próxima vez que se resuelva dicho subproblema.

Se aplican algoritmos de programación dinámica para resolver problemas de optimización; debido a que la programación dinámica garantiza hallar la solución óptima, al examinar todas las formas posibles de solucionar un problema y quedarse con la mejor solución.

**Hay dos propiedades importantes que un problema debe tener para poder aplicar la programación dinámica:**

- Subestructura óptima: significa que la solución dada a un problema de optimización, puede ser obtenida mediante la combinación de soluciones óptimas a sus subproblemas [19].
- Subproblemas superpuestos: significa que el espacio de los subproblemas debe de ser pequeño, es decir, que cualquier algoritmo recursivo que resuelva el problema debe resolver los mismos subproblemas una y otra vez, en lugar de generar nuevos subproblemas. La programación dinámica tiene en cuenta este hecho y resuelve cada subproblema una sola vez [19].

En ciencias de la computación, se dice que un problema que se puede descomponer recursivamente, tiene una subestructura óptima. Si los subproblemas se pueden anidar recursivamente dentro de problemas más grandes, se pueden aplicar los métodos de programación dinámica, habiendo una relación entre el valor de un problema mayor y los valores de los subproblemas [19].

En la literatura, a esta relación se la conoce como la ecuación de Bellman.

La ecuación de Bellman es una relación recursiva esencial que transcribe el principio de optimalidad de Bellman. La cuál se enuncia de la siguiente manera:

*“Una política óptima tiene la propiedad de que cualquiera que sea el estado inicial y la decisión inicial, las decisiones restantes deben constituir una política óptima con respecto del estado resultante de la primera decisión [3].”*

**Cuando se desarrolla un algoritmo de programación dinámica, seguimos una secuencia de cuatro pasos:**

1. Caracterizar la estructura de una solución óptima (afirmar que se satisface el principio de optimalidad).
2. Definir recursivamente el valor de una solución óptima.
3. Realizar el algoritmo que calcula el valor de una solución óptima, sin repetir cálculos de resultados de subproblemas.
4. Realizar un algoritmo para construir una solución óptima de la información computarizada anteriormente [25].

Si sólo queremos el valor de una solución óptima, y no la solución en sí, entonces no se realiza el paso 4. Cuando se realiza el paso 4, a veces se mantiene información adicional durante el paso 3 para que se pueda construir una solución óptima.

**Enfoques de la programación dinámica:**

- **Top-down (de arriba abajo):** El problema se divide en subproblemas; los cuales se resuelven de forma recursiva, memorizando o almacenando las soluciones de los subproblemas por si fuera necesario reutilizarlos más tarde.
- **Bottom-up (de abajo arriba):** Primero se resuelven los subproblemas y posteriormente se utilizan sus resultados para resolver los subproblemas más grandes.

## Capítulo 3

# Código de Cadena

El estudio de la representación de formas es un tema importante en la visión por computadora. Los códigos de cadena, [12], se utilizan para la representación de formas a partir del cambio en la dirección de los píxeles que forman cada objeto. Cada cambio de dirección en los píxeles corresponde a un elemento del código. Al final del recorrido de todo el objeto se tiene una secuencia de palabras del código que conforman la cadena asociados a números reales. Hay dos formas de conectar píxeles: cuatro conectividades y ocho conectividades.

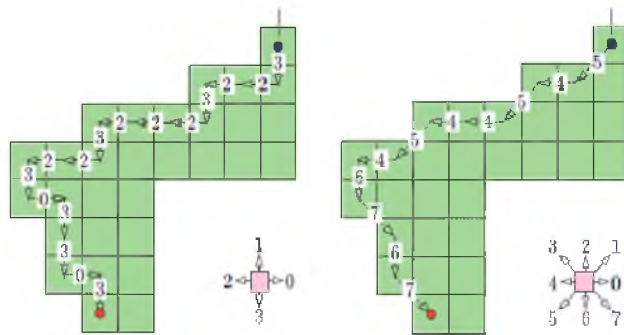


Figura 3.1: Código de cadena por dirección

En el artículo de McKee y Aggarwal [18], se ha utilizado el código de cadena en el proceso

de reconocimiento de objetos. Freeman [13] afirma que “en general, un esquema de codificación de estructuras de la línea debe satisfacer tres objetivos: (1) debe conservar fielmente la información de interés; (2) debe permitir un almacenamiento compacto y conveniente para la exhibición, y (3) debe facilitar cualquier procesamiento necesario”. El siguiente Código de Cadena por Vértice (VCC) cumple con estos tres objetivos.

### 3.1. Código de Cadena por Vértice (VCC)

Las principales características son:

1. Conserva la información y permite una considerable reducción de datos.
2. El código de cadena es invariante bajo traslación y rotación, y opcionalmente, en el punto de partida y de reflejo de la transformación.
3. Es válido utilizando formas triangulares, rectangulares y hexagonales en lugar de píxeles de cuadrados.
4. Los elementos de la cadena representan valores reales no símbolos.
5. Utilizando códigos de cadena es posible obtener relaciones entre el contorno de delimitación y el interior de la forma [12].

A continuación se presentan una serie de definiciones:

**Definición 1:** Un elemento  $a_i$  de una cadena indica el número de vértices, las cuales están en contacto con el contorno de delimitación de la forma en esa posición del elemento [12].

**Definición 2:** Una cadena  $A$  es una secuencia ordenada de elementos, y está representado por

$$A = a_1 \cdot a_2 \dots a_n = \{a_i : 1 \leq i \leq n\} [12]. \quad (1)$$

**Definición 3:** La longitud  $L$  de una cadena es la suma de las longitudes de sus elementos.  $L$  se puede expresar como

$$L = nl \quad (2)$$

Donde la longitud de cada lado de los pixeles se considera igual a uno [12].

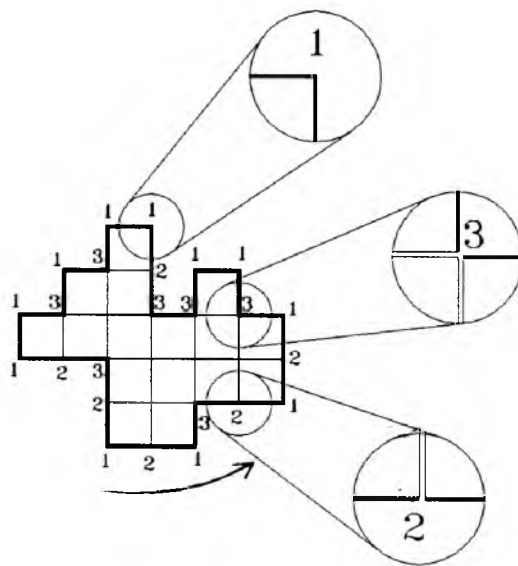


Figura 3.2: Código de cadena por vértice [12].

La fig. 2.12 muestra los elementos de la forma en píxeles; sólo hay tres diferentes números de vértices para el contorno de delimitación: 1, 2, y 3. Su cadena generada por VCC es:

121321213113321131311232

Y la longitud  $L$  de dicha cadena es 24.

## Capítulo 4

# Alineamiento de Secuencias

El alineamiento de secuencias es el proceso por el cual las secuencias son comparadas mediante la búsqueda de patrones de caracteres comunes y el establecimiento de la correspondencia entre residuos en las secuencias relacionadas.

El objetivo del alineamiento de secuencias por pares es hallar el mejor emparejamiento de dos secuencias, de modo que exista una correspondencia máxima entre los residuos. Para esto, una secuencia debe ser desplazada con respecto a la otra para hallar la posición donde se localizan las coincidencias máximas [19].

### 4.1. Alineamiento global y local

El alineamiento global considera la similitud en toda la extensión o longitud de las dos secuencias, para hallar el mejor alineamiento mediante el tamaño total entre las dos secuencias. Se utiliza más para alinear dos secuencias estrechamente relacionadas de aproximadamente igual tamaño.

El alineamiento local, no asume que las dos secuencias posean semejanza en toda la extensión; estas dos secuencias pueden ser de diferentes tamaños. Además el alineamiento local halla las regiones con el mayor nivel de similitud entre las dos secuencias y alinea esas regiones sin considerar el alineamiento del resto de las regiones de la secuencia.

### Ejemplo de alineamiento global y local.

Sea  $s_1 = \text{matematicas}$  y  $s_2 = \text{mmateaatcs}$ , la tabla 4.1 muestra los resultados de haber alineado  $s_1$  y  $s_2$  de manera global y local.

Alineamiento de secuencias global	Alineamiento de secuencias local
-matematicas	matemat
mmateaat-c-s	mateaat

Tabla 4.1: Ejemplo de alineamiento global y local.

## 4.2. Algoritmos de alineamiento

Los algoritmos de alineamiento global y local, pueden basarse en cualquiera de los tres métodos siguientes:

### 4.2.1. Método de la matriz de puntos

Es una forma gráfica de comparar dos secuencias en la matriz bidimensional. En una matriz de puntos, las dos secuencias a comparar son escritas en el eje horizontal y vertical de la matriz. La comparación se realiza mediante el escaneo de cada residuo de una secuencia por similitud con todos los residuos en la otra secuencia. Si una coincidencia residual es encontrada, un punto es colocado dentro del gráfico; de lo contrario, las posiciones de la matriz serian dejadas en blanco.

Cuando dos secuencias tienen regiones sustanciales de similitud, se forma una fila de puntos

para crear líneas contiguas y diagonales que revelan la secuencia de alineamiento. Si hay interrupciones en medio de una línea en diagonal, indica que hay inserciones o deleciones.

Existe un problema cuando se comparan largas secuencias utilizando el método de la matriz por puntos, concretamente el alto nivel de ruido. En la mayoría de los gráficos por puntos, los puntos son trazados sobre el gráfico, ocultando la identificación del alineamiento verdadero. Otra limitación de este método de análisis visual es que carece de rigor estadístico en la evaluación de la calidad del alineamiento [19].

#### 4.2.2. Método de programación dinámica

La programación dinámica es un método que encuentra el alineamiento óptimo al hacer coincidir dos secuencias para todos los posibles pares de caracteres entre las dos secuencias, basándose en el principio de optimalidad de Bellman.

Al igual que el método de matriz por puntos, se construye una matriz de alineamiento de dos dimensiones. Aunque, el alineamiento se halla de una forma más cuantitativa mediante la transformación de una matriz de puntos en una matriz de puntuación para mostrar las coincidencias y diferencias entre las dos secuencias. A veces se obtiene el alineamiento más preciso mediante la búsqueda del conjunto de puntuaciones más altas en dicha matriz.

##### 4.2.2.1. Traceback

Antes de mencionar cómo funciona el traceback indicaremos que los gaps son huecos que se le agregan a las secuencias (cadenas)  $A = a_1, a_2, \dots, a_n$  y  $B = b_1, b_2, \dots, b_m$ , cuando se avanza de manera vertical u horizontal en una matriz de sustitución o en una matriz de distancias mínimas  $M(n, m)$ . La notación de un hueco se hace por medio de un guión (-).

Los dos algoritmos de alineamientos de secuencias que se mencionarán posteriormente, alinean las secuencias a partir del traceback, el cual se construye de la siguiente forma:

- Si el avance es diagonal; es decir, si llegamos de la posición  $M(i-1, j-1)$ , no se le agregan gaps a las secuencias  $A$  y  $B$ . Como en los cuadros amarillos de la tabla 3.3.
- Si el avance es vertical; es decir, si llegamos de la posición  $M(i-1, j)$ , hay que añadir un gap a la secuencia  $A$ . Como en la línea rosada de la tabla 3.3.
- Si el avance es horizontal; es decir, si llegamos de la posición  $M(i, j-1)$ , hay que añadir un gap a la secuencia  $B$ . Como en la línea azul de la tabla 3.3.

**Ejemplo:**

Sea  $s_1 = matemáticas$  y  $s_2 = mmateatcs$ , aplicando el Algoritmo DTW tenemos la siguiente matriz de distancias mínimas.

	m	a	t	e	m	a	t	i	c	a	s
m	0	1	3	6	6	7	9	13	18	19	25
m	0	1	3	6	6	7	9	13	18	19	25
a	1	0	1	3	4	4	5	8	12	12	17
t	3	1	0	1	3	4	4	6	9	10	14
e	6	3	1	0	3	5	5	5	7	9	12
a	7	3	2	2	1	1	2	5	9	7	12

a	8	3	3	4	2	1	2	5	9	7	12
t	10	4	3	4	4	2	1	3	6	7	11
c	15	8	6	5	9	6	4	2	2	6	7
s	21	13	10	8	11	11	8	4	3	7	6

Tabla 4.2: Matriz de distancias mínimas de dos secuencias de palabras

Y el alineamiento de las secuencias  $s_1$  y  $s_2$  es:

m-aticas  
mmateaat-c-s

#### 4.2.2.2. Algoritmo de Needleman Wunsch

El algoritmo Needleman Wunsch encuentra regiones de similitud global que muestran la presencia de una relación entre dos secuencias [21].

Algoritmo:

Dadas dos secuencias  $A = a_1, a_2, \dots, a_n$  y  $B = b_1, b_2, \dots, b_m$  se construye una matriz  $M$  de  $n + 1$  filas y  $m + 1$  columnas. La secuencia  $B$  se ubica en las filas y la secuencia  $A$  en las columnas. Por medio de los siguientes pasos se calculan los valores de  $M$  que darán el puntaje de similitud entre  $A$  y  $B$ :

1. Se inicializa la matriz  $M$  o Matriz de Resultado con ceros; es decir,

$$M(i, 0) = M(0, j) = 0 \quad \text{con } 0 \leq i \leq n \text{ y } 0 \leq j \leq m \quad (1)$$

2. Calcular el valor de  $M(i, j)$  para  $\forall i \in [1, \dots, n]$  y  $\forall j \in [1, \dots, m]$  por medio de la Ecuación 2.

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + S(a_i + b_j) \\ M(i-1, j) + G_0 \\ M(i, j-1) + G_e \end{cases} \quad (2)$$

Donde

- $S(a_i, b_j)$  es una función de similitud (coincidencias o Matriz de Sustitución), que indica el puntaje dado por hacer coincidir a los elementos  $a_i$  y  $b_j$  de las secuencias a alinear.
- $G_k$  es el peso con el cual se penalizan los in/dels (inserciones o deleciones) de longitud  $k$ . Estos valores hacen referencia a los valores de los gaps (huecos), que son valores aleatorios.

3. Obtenida la matriz, sólo falta reconstruir el camino haciendo un traceback a partir de la posición  $M(m, n)$  ( que representa la puntuación más elevada entre las dos secuencias) hasta la posición  $M(0, 0)$  . Se avanza siempre hacia la diagonal izquierda o hacia arriba o hacia la izquierda, escogiendo siempre el valor mayor de las opciones precedentes. En caso de que el valor mayor sea el mismo en más de un vecino se le da prioridad a la diagonal izquierda [4].

#### 4.2.2.3. Algoritmo de Alineamiento Temporal Dinámico

Algoritmo de Alineamiento Temporal Dinámico (DTW) es un algoritmo que calcula un camino de alineamiento óptimo entre dos series de tiempo [26].

Algoritmo:

Dadas dos secuencias numéricas  $A = a_1, a_2, \dots, a_n$  y  $B = b_1, b_2, \dots, b_m$ . La longitud de las dos

secuencias puede ser diferente.

1. Se construye una matriz de distancias  $D_{n,m}$ , donde cada elemento  $D_{i,j}$  fue calculado por medio de la ecuación 1

$$D_{i,j} = |a_i - b_j|, 1 \leq i \leq n, 1 \leq j \leq m \quad (1)$$

Se pueden utilizar diferentes tipos de distancias; aunque la más utilizada es la distancia euclidiana.

2. Se construye una segunda matriz  $M_{n,m}$ , utilizando un algoritmo de programación dinámica y el siguiente criterio de optimización:

$$M_{i,j} = D_{i,j} + \min(M_{i-1,j-1}, M_{i-1,j}, M_{i,j-1})$$

Donde

- $M_{i,j}$  es la distancia mínima entre las subsecuencias  $a_1, a_2, \dots, a_i$  y  $b_1, b_2, \dots, b_j$ .

3. Un camino alineado es un camino a través de la matriz de distancia mínima del elemento  $M_{n,m}$  al elemento  $M_{1,1}$ , elemento constituido por aquellos elementos  $M_{i,j}$  que han formado la distancia  $M_{n,m}$ .

Hay tres condiciones impuestas en el algoritmo DTW que les aseguren una convergencia rápida:

1. Continuidad: el camino avanza poco a poco, paso a paso; los índices  $i$  y  $j$  se decrementan por máximo 1 unidad en un paso.
2. Frontera (Contorno):  $w_1 = (1, 1)$  y  $w_k = (m, n)$  son el comienzo y final del camino, donde  $k$  es la longitud del camino de deformación (alineamiento).

3. Monotonía: Tomando un elemento del camino  $w_h = (a, b)$  entonces  $w_{h-1} = (a', b')$  donde  $a \geq a'$  y  $b \geq b'$ . Obliga a que los puntos de  $w$  estén espaciados en el tiempo de forma continua [2].

Debido al principio de optimalidad la programación dinámica se aplica usando la técnica de "hacia atrás (backtracking)", la identificación del camino alineado utiliza un cierto tipo de estructura dinámica llamado "pila". Al igual que cualquier algoritmo de programación dinámica, el DTW tiene una complejidad polinomial.

#### 4.2.3. Método heurístico de palabra corta

El método heurístico de palabra corta busca trayectos cortos de letras iguales o aproximadamente iguales en dos secuencias. A las cadenas cortas de caracteres se les denomina 'palabras cortas', las cuales son parecidas a las ventanas que se utilizan en el método de matriz por puntos. Dicho método supone que al menos una palabra en común deben de tener dos secuencias relacionadas; primero identifica que la palabra coincida, para conseguir un alineamiento más grande mediante la extensión de las regiones de similitud de las palabras cortas. Cuando se encuentran las regiones con mayor similitud de secuencia, las regiones con mayor puntuación adyacentes se pueden unir en un alineamiento completo.

Existen dos algoritmos heurísticos para realizar búsqueda en bases de datos de gran magnitud; los cuales son BLASTA Y FASTA. Dichos algoritmos no garantizan encontrar el alineamiento óptimo, pero son de 50 a 100 veces más rápidos que la programación dinámica. En esta tesis no utilizaremos los algoritmos heurísticos, puesto que deseamos encontrar el mejor alineamiento posible.

## Capítulo 5

# Distancia propuesta

En esta sección se propone una nueva distancia que nos permite clasificar las cadenas resultantes de las formas en imágenes; por ejemplo, las escamas de las tortugas marinas. Dicha distancia se efectúa después de haber alineado dos cadenas, por medio del algoritmo de alineamiento temporal dinámico o el algoritmo de Needleman-Wunsch. Cabe mencionar que es recomendable alinear las cadenas con alguno de los algoritmos mencionados anteriormente, debido a que garantiza un alineamiento óptimo. La distancia que se propone es de gran utilidad en la clasificación de formas cuando se obtiene un buen alineamiento, como se muestra en los ejemplos del capítulo 6.

Sean  $c_1$  y  $c_2$  dos cadenas (secuencias) alineadas. Definamos su distancia de la siguiente manera:

$$d = \frac{h - |lc_1 - lc_2|/2 + nc}{lca}$$

Donde

- $nc$ = Son dígitos que están en la misma posición de las dos cadenas alineadas, pero son diferentes.
- $lc_1$ = Es la longitud de  $c_1$  antes de ser alineada.

- $lc_2$ = Es la longitud de  $c_2$  antes de ser alineada.
- $lca$ = Es la longitud de cualquiera de las dos cadenas alineadas.
- $h$ = Son todos los gaps que aparecen en las dos cadenas alineadas.

### **Demostración**

Sean  $s_1$ ,  $s_2$  y  $s_3$  cadenas alineadas, y sean  $ls_1$ ,  $ls_2$  y  $ls_3$  sus respectivas longitudes antes de ser alineadas. Mostraremos que  $d$  es una distancia; si cumple las siguientes propiedades:

1.  $d(s_1, s_2) \geq 0$ ,

Sea  $nc$  las no coincidencias y  $h$  los gaps entre  $s_1$  y  $s_2$ . Al observar  $d$  tenemos que  $nc \geq 0$  y  $h - |ls_1 - ls_2|/2 > 0$ , por lo tanto la distancia siempre es positiva.

2.  $d(s_1, s_2) = 0 \Leftrightarrow s_1 = s_2$

Si  $d(s_1, s_2) = 0$  entonces  $nc = 0$ ,  $h = 0$  y  $ls_1 = ls_2$ , lo cual implica que  $s_1 = s_2$ .

Si  $s_1 = s_2$  entonces  $nc = 0$ ,  $h = 0$  y  $ls_1 = ls_2$ , lo cual implica que  $d(s_1, s_2) = 0$ .

3.  $d(s_1, s_2) = d(s_2, s_1)$

Como la distancia propuesta penaliza los gaps de la misma forma en las dos cadenas y  $nc$  es el total de no coincidencias, se tiene la simetría.

4.  $d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3)$ ,

Si  $d(s_1, s_3) = 0$ ,  $d(s_1, s_2) = 0$ , y  $d(s_2, s_3) = 0$ , se obtiene el resultado deseado.

Si  $d(s_1, s_2) = 0$  utilizando la propiedad 2 tenemos que  $s_1 = s_2$ , lo cual implica que  $d(s_1, s_3) = d(s_2, s_3)$ . De manera análoga se cumple para  $d(s_1, s_3) = 0$  y  $d(s_2, s_3) = 0$ .

Si  $d(s_1, s_3) \neq 0$ ,  $d(s_1, s_2) \neq 0$ , y  $d(s_2, s_3) \neq 0$ . Mostramos que la desigualdad del triángulo se satisface de manera experimental, debido a que la distancia propuesta funciona sólo cuando es alineada de manera óptima; es decir, cuando se aplica algunos de los algoritmos mencionados. Se elaboró en Matlab el siguiente programa y se ejecutó 10000 veces, en las cuales

la desigualdad se cumple en un 99,9%.

```

1 SI=0;NO=0;s=0;n=0;contador=0;
2 while (contador<10000)
3   alin1=0; alin2=0; alin3=0; alin4=0; alin5=0; alin6=0; alin1_ATD1=0;
4   alin2_ATD1=0; alin1_ATD2=0; alin2_ATD2=0; alin1_ATD3=0; alin2_ATD3=0;
5   %crea secuencias de numeros
6   [s1,s2,s3]=crea_secuencia;
7   %Algoritmo alineamiento temporal dinamico
8   [alin1_ATD1, alin2_ATD1]=alin_temporal_dinamico(s1,s2);
9   [alin1_ATD2, alin2_ATD2]=alin_temporal_dinamico(s1,s3);
10  [alin1_ATD3, alin2_ATD3]=alin_temporal_dinamico(s2,s3);
11  %Distancias de las cadenas alineadas
12  D_12=inform_distancias(alin1_ATD1, alin2_ATD1,s1,s2);
13  D_13=inform_distancias(alin1_ATD2, alin2_ATD2,s1,s3);
14  D_23=inform_distancias(alin1_ATD3, alin2_ATD3,s2,s3);
15  %Verificacion de la desigualdad del triangulo
16  if ((D_13<=(D_12+D_23)) && (D_12<=(D_13+D_23)) && (D_23<=(D_12+D_13)))
17      SI=SI+1;
18  else
19      NO=NO+1;
20  end
21  %Algoritmo Needleman-Wunsch
22  [alin1, alin2]=Needleman_Wunsch(s1,s2);
23  [alin3, alin4]=Needleman_Wunsch(s1,s3);
24  [alin5, alin6]=Needleman_Wunsch(s2,s3);
25  %Distancias de las cadenas alineadas
26  D_12=inform_distancias(alin1, alin2,s1,s2);
27  D_13=inform_distancias(alin3, alin4,s1,s3);
28  D_23=inform_distancias(alin5, alin6,s2,s3);

```

```

29  %Verificacion de la desigualdad del triangulo
30  if ((D_13<=(D_12+D_23)) && (D_12<=(D_13+D_23)) && (D_23<=(D_12+D_13)))
31      s=s+1;
32  else
33      n=n+1;
34  end
35  contador=contador+1;
36  end

```

**Nota:**

Unas de las secuencias en la que no se cumple la desigualdad del triángulo son  $s_1 = 21221111323$ ,  $s_2 = 1313132221$  y  $s_3 = 211212211132211$ . La tabla 5.1 muestra los alineamientos con el algoritmo de Needleman-Wunsch; y sus respectivas distancias:

Secuencias	Alineamiento	Distancia
$s_1$	21221111323	0.68182
$s_2$	-1313132221	
$s_1$	2-12-211113-2-3	0.26667
$s_3$	211212211132211	
$s_2$	—1-313132221	0.36667
$s_3$	211212211132211	

Tabla 5.1: Ejemplo de alineamiento de secuencias con el algoritmo de Needleman-Wunsch, y sus respectivas distancias

Al sustituir las distancias de la tabla 5.1 en la desigualdad del triángulo  $D(s_1, s_2) \leq D(s_1, s_3) + D(s_2, s_3)$ , notamos que esta no se cumple.

La distancia entre  $s_1$  y  $s_3$  es muy pequeña, lo cual significa que las cadenas son similares. De

manera análoga sucede con  $s_2$  y  $s_3$ . Pero la distancia entre la secuencia  $s_1$  y  $s_2$  es grande, lo cual significa que las cadenas varían. La desigualdad del triángulo falló porque en secuencias de longitud pequeña, que dos cadenas sean similares a otra, no siempre da como resultado que ellas sean iguales entre sí. Además la desigualdad del triángulo falla cuando el alineamiento no es el más óptimo en secuencias pequeñas.

Cabe mencionar que la desigualdad del triángulo si se cumplió al alinear las secuencias  $s_1, s_2$  y  $s_3$  con el algoritmo de alineamiento temporal dinámico. Lo cual sucede debido a que el alineamiento si fue óptimo.

## Capítulo 6

# Protocolo experimental

### 6.1. Preprocesamiento de las imágenes

Para clasificar de manera exitosa una imagen primero se debe realizar un preprocesamiento (segmentación), en el cual ciertas características de una imagen se resaltan (por ejemplo los bordes) y otras se eliminan (por ejemplo el ruido).

Mediante la segmentación vamos a dividir la imagen en las partes u objetos que la forman. El nivel al que se realiza esta subdivisión depende de la aplicación en particular, es decir, la segmentación terminará cuando se hayan detectado todos los objetos de interés para la aplicación. En general, la segmentación automática es una de las tareas más complicadas dentro del procesado de imagen. La segmentación va a dar lugar en última instancia al éxito o fallo el proceso de análisis. En la mayor parte de los casos, una buena segmentación dará lugar a una solución correcta, por lo que, se debe poner todo el esfuerzo posible en la etapa de segmentación.

Uno de los métodos en la segmentación de imágenes es la umbralización. La cual consiste en buscar un valor (umbral) adecuado entre los valores de grises en la imagen que permita separar de manera apropiada el fondo y el objeto; es decir, que permita binarizar a la imagen de forma

correcta. Diversos métodos de umbralización están fundamentados en la información estadística que ofrece el histograma.

En cada imagen que inserta el usuario al programa se obtienen dos umbrales diferentes, los cuales se elaboraron de la siguiente manera:

1. Se obtuvo la media y la mediana del histograma de la imagen, posteriormente se sumaron las posiciones dentro del histograma de la media y la mediana, por último se dividió entre dos. De la manera anterior se obtuvo el umbral 1.
2. Al primer umbral se le suma la posición del mínimo dentro del histograma y los divide entre dos. Así se obtiene el umbral 2.

El programa solo binariza con uno de estos dos umbrales obtenidos de cada imagen, elige el más adecuado por medio de una condición, la cual consiste de valores tomados del histograma.

La condición es la siguiente:

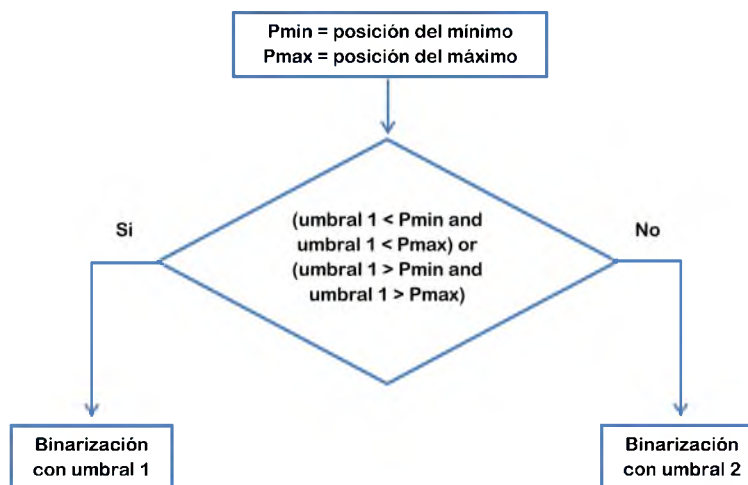


Figura 6.1: Diagrama de flujo de la condición que se utiliza para binarizar. (Nota: Todos los valores son adquiridos del histograma)

### 6.1.1. Prueba 1 de la binarización

Analizando los valores de la tabla, podemos deducir que se binarizó utilizando el umbral 1; ya que las posiciones del máximo y mínimo del histograma de la imagen son menores que el umbral 1, y esto cumple la condición de la figura 6.1.

Umbral 1	120.7834
Posición del Mínimo	15
Posición del Máximo	73
Umbral 2	67.8917

Tabla 6.1: Valores que utiliza el programa para binarizar y en la condición de binarización

Al comparar las siguientes dos imágenes binarizadas con los dos umbrales mencionado anteriormente, podemos percibir que efectivamente la condición si se queda con la mejor binarización.



Figura 6.2: Binarización con umbral 1



Figura 6.3: Binarización con umbral 2

### 6.1.2. Prueba 2 de la binarización

Al analizar los valores de la tabla y la condición de la figura 6.1, concluimos que se utilizó el umbral 2 para binarizar la imagen; debido que la posición del máximo del histograma de la imagen

es menor que el umbral 1 y la posición del mínimo del histograma de la imagen es mayor que el umbral 1.

Umbral 1	109.0513
Posición del Mínimo	215
Posición del Máximo	42
Umbral 2	162.0257

Tabla 6.2: Valores que utiliza el programa para binarizar y en la condición de binarización

Al observar las siguientes dos imágenes binarizadas, comprobamos que la condición es confiable.



Figura 6.4: Binarización con umbral 1

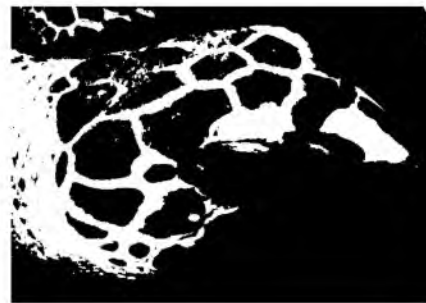
























Figura 6.5: Binarización con umbral 2

## 6.2. Distancias de Cadenas alineadas de polígonos

Se binarizaron las imágenes (tabla 6.3) de manera general y se generaron sus cadenas, utilizando códigos de cadena por vértice debido a que es invariante bajo rotación. Posteriormente se alinearon las cadenas de los diferentes polígonos con la cadena del círculo, y se calcularon sus respectivas distancias. Los resultados fueron:

Polígonos	Círculo	Alineamiento temporal dinámico	Needleman-Wunsch
		0.657	0.622
		0.485	0.466
		0.431	0.396
		0.513	0.366
		0.397	0.339
		0.353	0.294
		0.321	0.265
		0.335	0.257
		0.280	0.239
		0.274	0.202
		0.209	0.186





		0.187	0.159
		0.142	0.154

Tabla 6.3: Distancias resultantes de las cadenas alineadas (de los diferentes polígonos) por los algoritmos ATD y Needleman-Wunsch.

En la gráfica 6.6 se puede observar que conforme incrementan los lados del polígono la distancia disminuye, lo cual sucede debido a que un polígono con lados infinitos es un círculo; es decir, al aumentar los lados del polígono este se va pareciendo cada vez más a un círculo.

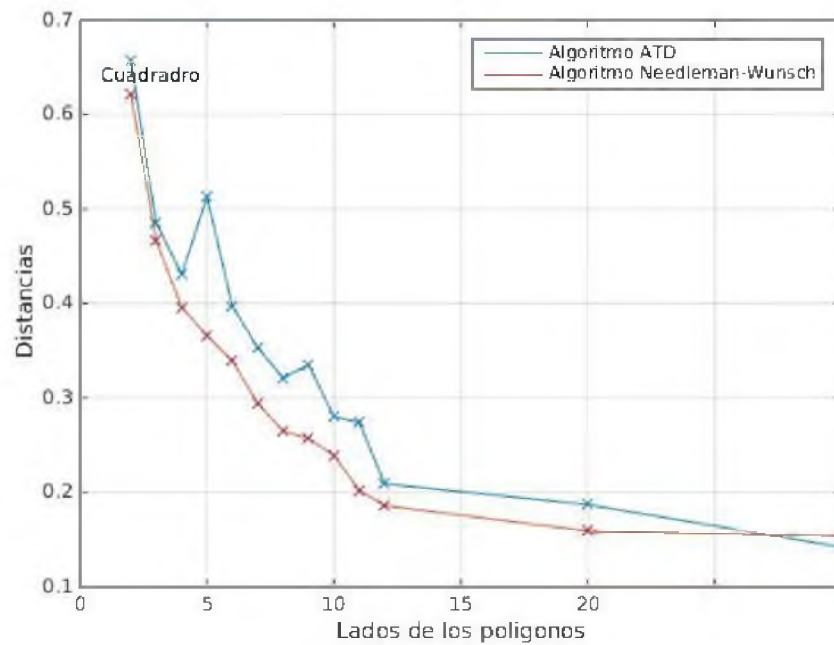


Figura 6.6: Representación gráfica de distancias resultantes de las cadenas alineadas.

En la gráfica anterior, hay un incremento en la distancia del pentágono alineado por el algoritmo ATD, esto debido a que el polígono anterior es un rombo, por lo que su cadena está formada en su mayoría por unos y tres, de manera similar ocurre en la cadena del círculo; la cadena del pentágono contiene aproximadamente un quinto de número dos, es por ello que sucede en este caso dicho incremento.

Del análisis anterior, logramos concluir que la distancia propuesta es efectiva para clasificar pares de cadenas alineadas de polígonos. Se analizaron las imágenes de polígonos; debido que las escamas de las tortugas son polígonos irregulares, y en este trabajo se clasificaron las imágenes de tortugas marinas, usando algunas de sus escamas distintivas.

### 6.3. Comparación entre la tortuga Verde y Carey

Existen diversas maneras de identificar a las diferentes especies de tortugas marinas; en este trabajo se clasificaron por medio de dos de sus escamas (figura 6.7 y 6.8), debido a que presentan una similitud entre especies y variación con respecto a las otras especies de tortugas.

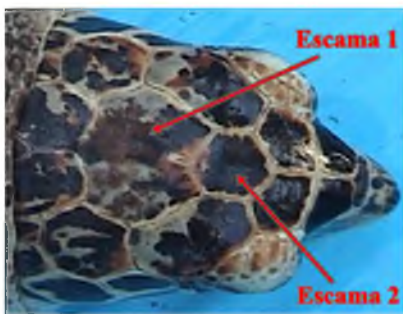


Figura 6.7: Escamas 1 y 2 de la tortuga carey    Figura 6.8: Escamas 1 y 2 de la tortuga verde

### 6.3.1. Comparación de las escamas 1 de las tortugas carey y verde

La prueba consiste en comparar las imágenes de las tortugas carey y verde, utilizando la escama 1. Cabe mencionar que se recortó la zona de interés (escamas 1) en las imágenes de las dos especies.



Figura 6.9: Ejemplo de escama 1 de la tortuga carey



Figura 6.10: Ejemplo de escama 1 de la tortuga verde

La figura 6.9 y 6.10 son ejemplos de las formas de las escamas 1 en las tortugas carey y verde, respectivamente.



Figura 6.11: Binarización de la figura 6.9



Figura 6.12: Binarización de la figura 6.10

El código de cadena de la figura 6.11 es:

```
s1 = 1 2 2 2 2 2 1 2 3 2 2 2 2 1 3 2 2 1 3 2 2 1 3 2 2 1 3 3 1 3 1 2 2 3 2 2 1 3 2 2 1 2 3 3 1 3 2
2 2 2 1 3 2 2 2 1 3 3 1 3 3 1 3 3 1 3 2 1 2 3 1 3 1 3 2 2 1 3 1 3 1 3 1 3 1 2 2 2 2 1 3 3 1 3 1 2 3
2 2 1 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 2 2 2 2 2 3 1 2 3 1 3 1 2 3 1 3 1 2 3 1 3 1 2
3 1 2 3 2 1 2 2 2 2 2 1 3 1 2 2 2 2 2 3 1 2 3 1 3 1 2 3 1 3 1 2 3 1 2 2 2 3 1 2 3 2 2 2 1 3 1
```

3 1 2 3 1 3 2 1 2 3 1 2 2 3 1 2 2 3 1 2 2 3 1 2 3 1 2 2 3 1 2 2 2 2 2 2 2 1 3 3 1 3 3 1 3 3  
 1 3 3 1 3 2 2 1 3 2 2 2 2 1 3 2 2 1 3 2 2 1 3 1 3 3 1 3 2 2 1 3 2 1 2 3 2 2 2 1 3 3 1 3 1 3 1 2 2 2  
 2 1 3 3 1 3 3 1 3 1 3 2 2 2 2 3 1 2 2 3 1 2 2 3 1 2 3 1 3 1 2 3 1 2 3 1 2 3 1 3 1 3 1 2 3 1 3 1 2  
 2 2 2 2 2 1 3 1 3 1 2 3 1 3 1 3 1 3 1 3 3 1 3 1 3 1 3 3 1 3 3 1 3 3 1 3 2 2 2 1 3 2 2 2 2 2 2 2 3  
 2 2 1 3 1 2 2 2 2 1 2 2 2 2 3 1 2 2 2 3 1 2 2 2 3 1 2 2 2 3 1 2 2 2 2 3 1 2 3 1 2 2 3 1 3 1 2 3

El código de cadena de la figura 6.12 es:

s2 = 1 2 2 2 2 2 2 2 2 2 2 2 1 3 1 3 1 3 1 3 1 3 1 2 3 1 3 1 3 1 3 1 3 1 2 3 1 3 1 2 3 3 1 3 1 3 1 3  
 1 3 1 2 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 2 3 1 2 3 1 3 1 2 2 2 3 1 2 2 2 2 2 2 2 2 2 2 2 2  
 2 2 1 3 2  
 2 3 1 3 1 3 3 1 3 1 3 1 3 1 3 1 3 1 3 1 2 3 1 3 1 3 1 3 1 3 1 3 2 1 2 2 3 1 2 3 1 3 1 3 1 3 1 2 3 1 2  
 3 1 2 2 2 3 1 2 3 1 2 2 2 2 2 2 2 2 1 3 2 2 2 2 2 2 2 1 3 2 2 2 1 3 2 2 2 2 1 3 1 3 1 3 1 3 1 3 1 2  
 2 3 1 3 1 3 1 2 3 1 3 1 3 1 2 2 3 1 3 1 3 1 2 3 1 2 2 3 1 2 3 1 2 2 3 1 2 3 1 2 3 1 2 3 1 3 1 3 1  
 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 3 1 3 1 2 3 1 3 2 1 2 3 1 2 2 2 2 2 2 2 1 3 1 3 3 1 3 1 3 1 3 1 3 1 3  
 1 2 3 1 3 1 3 1 3 1 3 1 3 1 3 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 3 1 3 1 3 3 1 3 3 1 3 1 3 1 3 1 3 1  
 3 1 3 2 2 1 3 3 1 3 1 3 1 3 1 3 1 3 3 1 3 3 1 3 1 3 1 3 1 3 1 3 1 2 3 1 2 3 1 3 1 2 2 2 3 1 2 2 2 2 2 2 3

El alineamiento de las dos cadenas anteriores con el algoritmo de alineamiento temporal dinámico es:

```

1 |alin1_ATD =
2 |12-----22221-23222213-2213221322133131223---221322123313222213-22213313313313
3 |-2-12313132-----213-----131313-1222
4 |213313-1-2322131331331331331331331331322222312313123131231-312--3123-2-----1
5 |-2--222213122222312-31312313123122231-23222131312-313212312231223122312-31
6 |2231231222222221331331331331322132222132213221313313-22132123-2---2213313-131
7 |2222133133131322223122312231231312312313-13123131222222131312313-1313-13313
8 |13133133133132221322222232213-12222122223-12223122312223122223122222312312--
9 |231312-----3
    
```



La distancia de las cadenas alineadas por el algoritmo de alineamiento temporal dinámico es 0.47217. Y la distancia de alin1 y alin2 es 0.40167.

De las distancias obtenidas al analizar todas las imágenes de muestra (escamas 1), tenemos las siguientes medias:

Comparación	Media de las distancias	
	Alineamiento temporal dinámico	Needleman-Wunsch
Carey-carey	0.36	0.29
Verde-verde	0.39	0.32
Carey-verde	0.48	0.40

Tabla 6.4: Medias de las distancias obtenidas con el análisis de las escamas 1 de las tortugas carey y verde.

Al analizar las distancias medias de la tabla 6.4, tenemos que la distancia obtenida de las cadenas alineadas por el algoritmo de Alineamiento Temporal Dinámico se aproxima más a la media que pertenece a las distancias obtenidas de las cadenas alineadas por el algoritmo de Alineamiento Temporal Dinámico entre las tortugas carey y verde. De manera análoga sucede con la distancia obtenida de las cadenas alineadas por el algoritmo de Needleman-Wunsch. De esta manera, concluimos que las escamas efectivamente son de dos especies diferentes de tortugas marinas, en este caso una pertenece a la carey y otra a la verde.

Las líneas negras (figura 6.13 y 6.14) nos muestran las distancias obtenidas de las escamas de tortugas carey al ser comparada con la figura 6.9. Y las otras líneas son las distancias obtenidas al comparar las escamas de las tortugas carey con las escamas de las tortugas verdes.

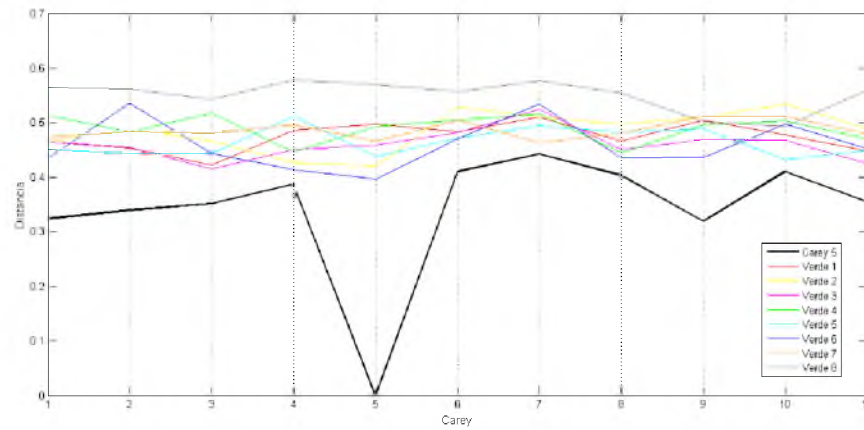


Figura 6.13: Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.9.

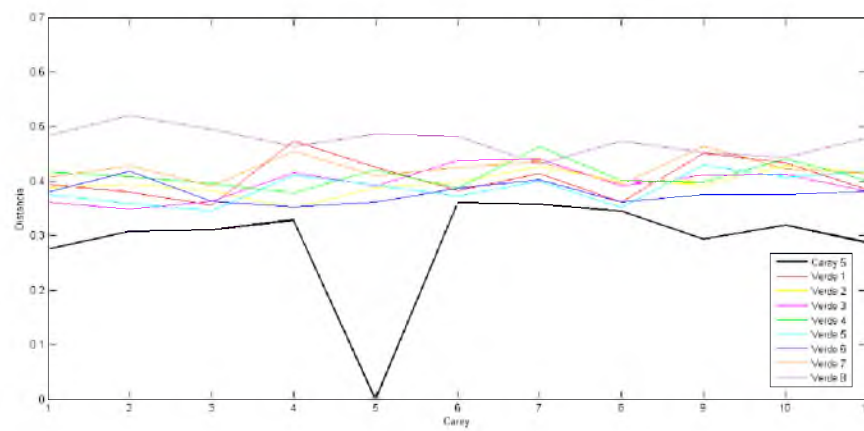


Figura 6.14: Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.9.

Las líneas negras (figura 6.15 y 6.16) nos muestran las distancias obtenidas de las escamas de tortugas verde al ser comparada con la figura 6.10. Y las otras líneas son las distancias obtenidas al comparar las escamas de las tortugas carey con las escamas de las tortugas verdes.

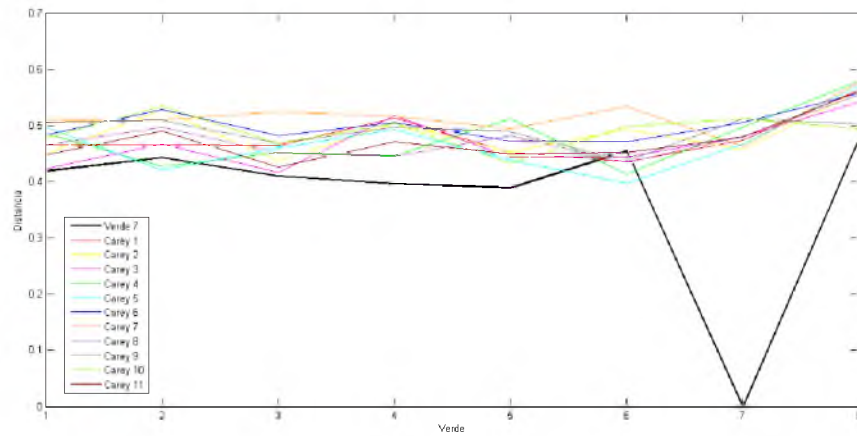


Figura 6.15: Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.10.

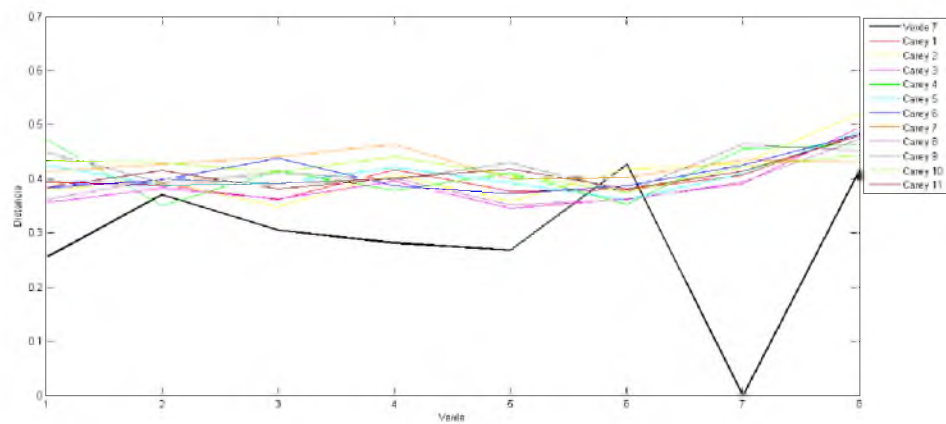


Figura 6.16: Gráfica de distancias obtenidas de las escamas 1 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.10.

Las figuras 6.15 y 6.16 muestran que no siempre se va a obtener el resultado deseado; esto debido a que algunas imágenes tienen demasiado ruido; es decir, la forma de la escama que mejor define a cada especie está distorsionada.









las siguientes medias:

Comparación	Media de las distancias	
	Alineamiento temporal dinámico	Needleman-Wunsch
Carey-carey	0.35	0.25
Verde-verde	0.42	0.27
Carey-verde	0.49	0.33

Tabla 6.5: Medias de las distancias obtenidas con el análisis de las escamas 2 de las tortugas carey y verde.

Las líneas negras (figura 6.21 y 6.22) nos muestran las distancias obtenidas de las escamas de tortugas carey al ser comparada con la figura 6.17. Y las otras líneas son las distancias obtenidas al comparar las escamas de las tortugas carey con las escamas de las tortugas verdes.

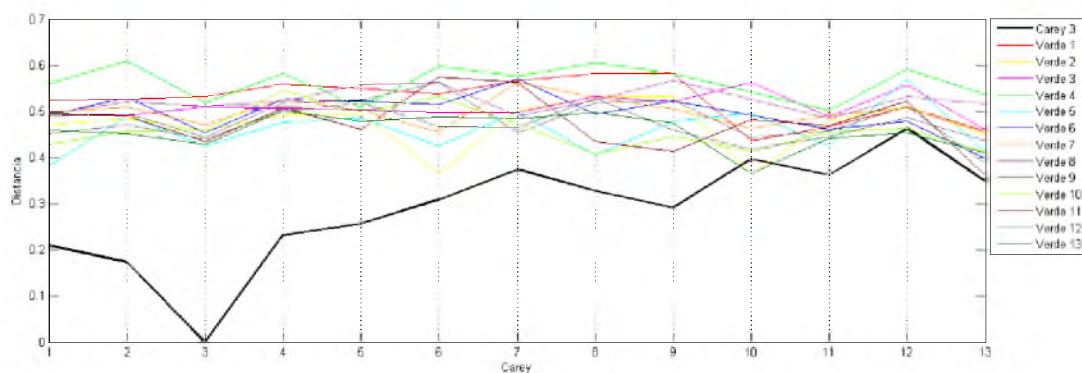


Figura 6.21: Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.17.

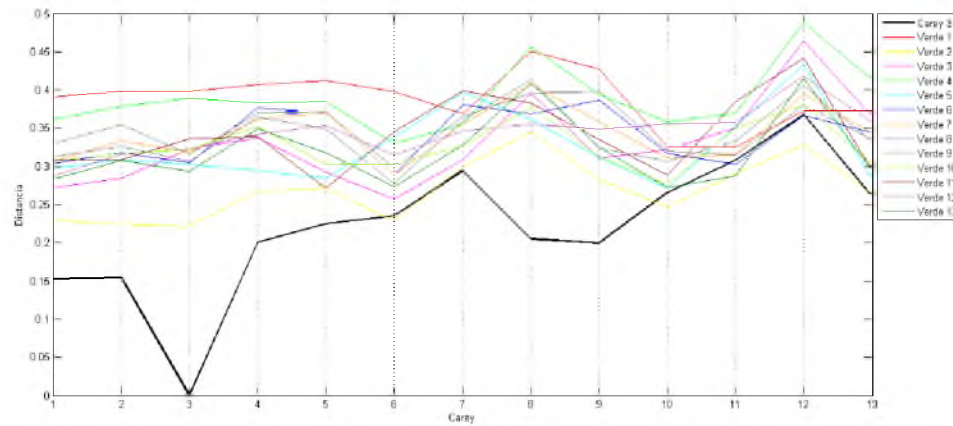


Figura 6.22: Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.17.

Las líneas negras (figura 6.23 y 6.24) nos muestran las distancias obtenidas de las escamas de tortugas verde al ser comparada con la figura 6.18. Y las otras líneas son las distancias obtenidas al comparar las escamas de las tortugas carey con las escamas de las tortugas verdes.

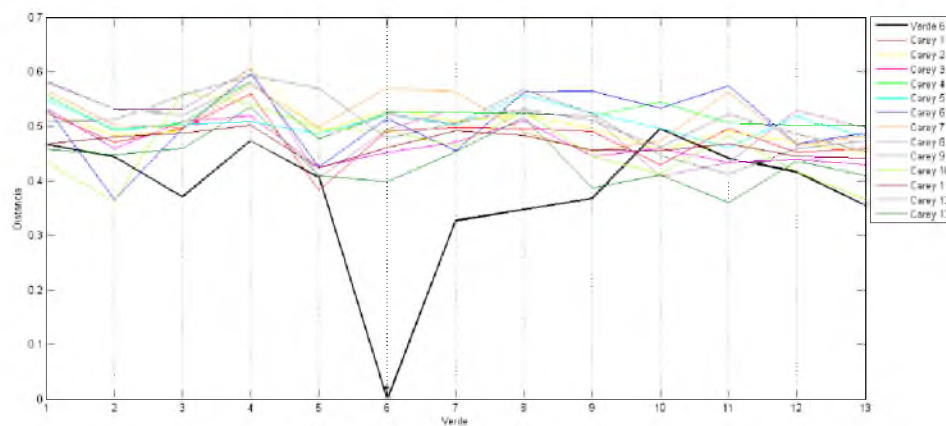


Figura 6.23: Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo de alineamiento temporal dinámico), al analizar la figura 6.18.

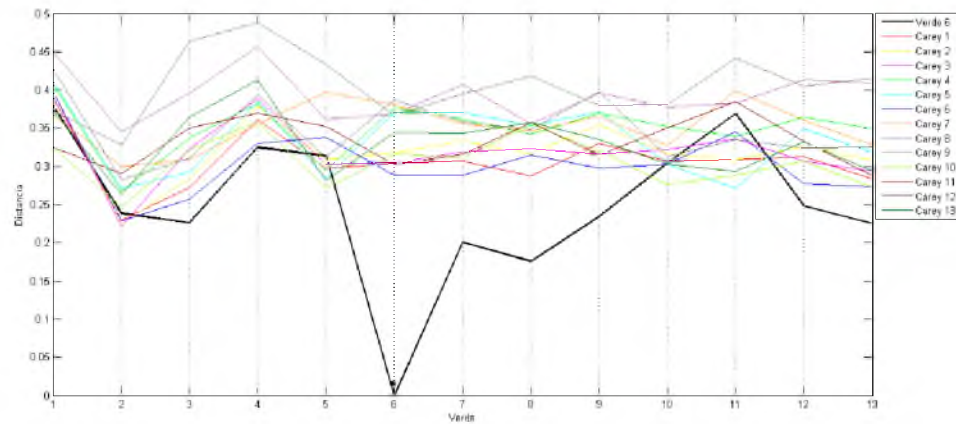


Figura 6.24: Gráfica de distancias obtenidas de las escamas 2 (utilizando el algoritmo Needleman-Wunsch), al analizar la figura 6.18.

En las gráficas 6.21, 6.22, 6.23 y 6.24 se muestran ciertos casos en los cuales no se puede percibir el tipo de especies al que pertenecen las escamas, lo cual sucede porque existen imágenes de escamas que presentan algunas deformaciones.

## Capítulo 7

# Conclusiones

La frase *"la matemática es el alfabeto con el que Dios escribió el universo"* de Galileo Galilei para mí tiene mucha razón, porque en este trabajo de tesis como en muchos otros proyectos la matemática ha desarrollado un papel muy importante en la obtención de soluciones. En este trabajo se propuso una distancia basada en los códigos de cadena que nos permitió clasificar imágenes de tortugas marinas.

Para realizar un análisis de imágenes, primero es necesario investigar sobre el objeto, fenómeno o ser vivo que se desea identificar y clasificar. En esta tesis se investigó sobre la morfología de las tortugas marinas, en particular sobre las tortugas carey y verde, ya que son unas de las especies que se encuentran en peligro de extinción. Existen diversas formas de identificarlas, sin embargo nosotros lo hicimos utilizando dos de las escamas que diferencian a las dos especies entre sí.

Cabe mencionar que la computación al igual que las matemáticas son de gran utilidad en el análisis de imágenes. Se utilizó el lenguaje de programación Matlab, para crear dos programas que nos ayudaran en el análisis de las imágenes. El primer programa binariza las escamas de las tortugas, y posteriormente obtiene el código de cadena correspondiente a cada imagen. El segundo programa alinea dos pares de cadenas utilizando los algoritmos de alineamiento temporal dinámi-

co y Needleman-Wunsch. Una vez alineadas las cadenas se obtiene su distancia con la cual se decide si las imágenes comparadas son o no de la misma especie.

El algoritmo de Needleman-Wunsch tiene como desventaja que es poco apto al alinear cadenas de diferentes longitudes; es decir, longitudes muy variadas. En general, las distancias obtenidas al alinear las cadenas por los algoritmos de alineamiento temporal dinámico y Needleman-Wunsch son eficientes al clasificar las formas características de las imágenes.

Se probó en algunas imágenes la eficiencia de los programas. Aunque no contamos con imágenes suficientes y de buena calidad de tortugas carey y verde, podemos decir que la distancia si es eficiente para obtener una adecuada clasificación.

# Bibliografía

- [1] Ana González Marcos, Francisco Javier Martínez de Pisón Ascacibar, Alpha Verónica Pernía Espinoza, Fernando Alba Elías, Manuel Castejón Limas, Joaquín Ordieres Meré y Eliseo Vergara González (Integrantes del Grupo de Investigación EDMANS), 2006. Técnicas y algoritmos básicos de visión artificial. Universidad de la Rioja, Servicio de Publicaciones, España.
- [2] Antonio Moreno García, 2012. Bases de Datos de Series Temporales: Representación y Consultas. Trabajo fin de máster; Universidad de Castilla-La Mancha. Escuela Superior de Ingeniería Informática. Departamento de Sistemas Informáticos, Programa Oficial de Postgrado en Tecnologías Informáticas Avanzadas.
- [3] Richard E. Bellman and Stuart E. Dreyfus, 1962. Applied Dynamic Programming. Princeton University Press; London: Oxford University Press.
- [4] Carles Pons Noguera, 2010. Alineamiento de Secuencias Genéticas en Procesadores MultiCore. Universidad Autónoma de Barcelona.
- [5] Daniel Galavís Ibáñez, Edgar Jiménez Suárez, 2005. Implementación del algoritmo Smith y Waterman para el análisis de alineamientos de secuencias proteómicas. Tesis de Ingeniería, Corporación Universitaria Rafael Núñez, Cartagena de Indias.

- [6] Daniel Malacara "Óptica Tradicional y Moderna" [Página web en línea]. La Ciencia para Todos, (1997). México, D.F. Disponible: [http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec\\_9.htm](http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec_9.htm)
- [7] Eckert, Karen L. y Jennifer Beggs. 2006. Mercado de Tortugas Marinas. Un Manual de Métodos Recomendados. Red de Conservación de Tortugas Marinas del Gran Caribe (WIDECAS) Informe Técnico No. 2. Edición Revisada. Beaufort, North Carolina USA. 40 pp.
- [8] Eckert, K. L., K. A. Bjorndal, F. A. Abreu-Grobois y M. Donnelly (Editores). 2000 (Traducción al español). Técnicas de Investigación y Manejo para la Conservación de las Tortugas Marinas. Grupo Especialista en Tortugas Marinas UICN/CSE Publicación No. 4.
- [9] Enzo Rucci, 2013. Computación eficiente del alineamiento de secuencias de ADN sobre cluster de multicores. Universidad Nacional de La Plata.
- [10] Rucci Enzo, De Giusti Armando E., Chichizola Franco, Naiouf R. Marcelo, De Giusti Laura C. (2010). Comparación de modelos de comunicación/sincronización en Programación Paralela sobre Cluster de Multicores. Instituto de Investigación en Informática LIDI (III-LIDI); Facultad de Informática Universidad Nacional de La Plata, Argentina.
- [11] Elizabeth Labastida Estrada, Foto-identificación de tortuga carey (*Eretmochelys imbricata*) y blanca (*Chelonia mydas*) en las playas de Chenkan e Isla Aguada, Campeche. Tesis de Licenciatura, Universidad Autónoma Metropolitana Unidad Xochimilco.
- [12] Ernesto Bribiesca (1998). A new chain code. *Pattern Recognition* 32 (1999) 235-251.
- [13] H. Freeman, Computer processing of line drawing images, *ACM Comput. Surveys* 6 (1974) 57-97.

- [14] Jean C., S Ciccione, E. Talma, K. Ballorain y J. Bourjea. 2010. Photo-identification method for green and hawksbill turtles. First results from Reunion. Indian Ocean Turtle Newsletter No. 11.
- [15] Jeanette Wyneken, 2004. La Anatomía de las Tortugas Marinas. U.S. Department of Commerce NOAA Technical Memorandum NMFS-SEFSC-470,172 pp. [Versión en español de Wyneken, J. 2001. The Anatomy of Sea Turtles. U.S. Department of Commerce NOAA Technical Memorandum NMFS-SEFSC, 172 pp.]
- [16] Jorge Lira Chávez, 2002. Introducción al tratamiento digital de imágenes (Primera edición). Instituto Politécnico Nacional, Universidad Nacional Autónoma de México, Fondo de Cultura Económica México. México, D.F.
- [17] Jorge Lira Chávez, 2010. Tratamiento digital de imágenes multiespectrales (Segunda edición). Instituto de Geofísica, Universidad Nacional Autónoma de México. México, D.F.
- [18] J.W. McKee, J.K. Aggarwal, Computer recognition of partial views of curved objects, IEEE Trans. Comput. C-26 (1977) 790-800.
- [19] Óscar Sánchez Becerro, 2014. Algoritmos de programación dinámica con R para resolver problemas de alineamiento de secuencias. Trabajo fin de máster; Universidad Complutense de Madrid (UCM).
- [20] Pro. J. Kittler, 2002. Notas del seminario de Reconocimiento de Patrones de Grupo de Tratamiento de Imágenes del Instituto de Ingeniería Eléctrica. Universidad de Surrey.
- [21] Saul B. Needleman y Christian D. Wunsch, 1970. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. Department of Biochemistry, Northwestern University, and hlwlear Medicine Service, V. A . Research Hospital Chicago, Ill. 60611, U.S.A.

- [22] Scale Invariant Feature Transform (SIFT) [Página web en línea]. Disponible: <http://robwhess.github.io/opensift/>
- [23] Secretaría CIT (2004). Una Introducción a las Especies de Tortugas Marinas del Mundo. Octubre 2004, San José, Costa Rica.
- [24] Significados [Página web en línea]. Disponible: <http://www.significados.com/morfologia/>
- [25] Thomas H.Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (third edition).
- [26] Titus Felix Furtuna, 2008. Dynamic Programming Algorithms in Speech Recognition. Academy of Economic Studies, Bucharest.
- [27] Unión Interamericana para la Conservación de la Naturaleza (UICN) [Página web en línea]. Disponible: <http://www.iucn.org/es/>
- [28] Zeeuw P., E. Pauwels, E. Ranguelova, D. Buonantony, S. Eckert, 2010. Computer Assisted Photo Identification of Dermochelys coriacea. Proceedings of International Conference on Pattern Recognition (ICPR).