



UNIVERSIDAD MICHOACANA DE SAN
NICOLÁS DE HIDALGO

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
"MAT. LUIS MANUEL RIVERA GUTIÉRREZ "

PREDICCIÓN DE LA POSICIÓN DE FUENTES DE
REPRODUCCIÓN DE MOSQUITOS CON
APLICACIONES ELECTRÓNICAS Y SIMULACIONES
NUMÉRICAS

T E S I S

EN CUMPLIMIENTO DE LOS REQUISITOS PARA OBTENER
EL GRADO DE:

MAESTRO EN CIENCIAS EN INGENIERÍA FÍSICA

PRESENTA:

LIC. EDUARDO MELLADO VILLASEÑOR



DIRECTOR DE TESIS:

DR. JOSÉ ANTONIO GONZÁLEZ CERVERA

MORELIA MICH.

SEPTIEMBRE DE 2020

Para todos mis seres cercanos...

Índice general

Agradecimientos	V
Resumen	VI
Abstract	VIII
1. Introducción	1
1.1. Componentes electrónicos	3
1.1.1. Arduino Due	4
1.1.2. Módulo DHT22	6
1.1.3. Módulo Bluetooth HC-06	8
1.2. Modelos matemáticos	9
1.3. Inteligencia artificial	11
1.4. Trabajo preliminar	12
2. Elementos electrónicos	14
2.1. Arduino Due	14
2.2. Atmel Studio 7	16
2.3. Configuración del módulo DHT22	22

2.4. Configuración del módulo HC-06	23
2.5. Lámpara de LEDs Ultravioleta	24
3. Métodos numéricos	26
3.1. Métodos introductorios	26
3.1.1. Método de Euler	27
3.1.2. Métodos Runge Kutta	33
3.2. Diferencias finitas	44
3.3. Redes Neuronales Artificiales	53
3.3.1. Forma de aprendizaje	53
3.3.2. Redes neuronales como un modelo del cerebro humano	54
3.3.3. Topología de una red neuronal	55
3.3.4. Red neuronal generalizada	56
4. Aplicación a la propagación de mosquitos	64
4.1. Ecuación de reacción difusión	64
4.2. Construcción del dispositivo electrónico	69
4.2.1. Etapa uno	70
4.2.2. Etapa dos	70
4.2.3. Etapa tres	71
4.2.4. Validación	72
4.3. Gráfica de datos	73
4.4. Procesamiento de imagen	85

5. Discusión y Conclusiones	93
5.1. Dispositivo electrónico	93
5.2. Ecuación de reacción-difusión	95
5.3. Perspectiva	96
Apéndices	98
A. Sistema de numeración	98
A.1. Conversiones	99
A.2. Código ASCII	104
Bibliografía	106

Agradecimientos

Durante el desarrollo de esta tesis he conocido personas increíbles que han hecho que el proyecto que estoy desarrollando sea posible. A estas personas los admiro mucho ya que me han enseñado el camino del bien, por así decirlo, ellos me instruyeron para seguir el camino de un investigador. Quiero darle las más sinceras gracias. Mi madre Blanca la que a pesar de su edad este año logró titularse de la escuela normal superior dejando un claro ejemplo de que se puede hacer cosas sin importar la edad que tengas, además de demostrar su gran fortaleza. Finalmente a mi amiga Lety que me apoyó y me dio grandes consejos.

Morelia MEX., 31 de agosto de 2020

Resumen

El mosquito a sido considerado como el animal más letal del mundo debido al número de muertes que causa por la transmisión de enfermedades a través de su picadura hacia los seres vivos y que en su mayoría no se cuenta con una vacuna como tal, ya que si el ser vivo infectado no se atiende a tiempo, este tiende a morir. Conocer la posición de las fuentes de reproducción de los mosquitos nos permite realizar trabajos para fumigar dicha área y no contaminar otras áreas donde no se requiera, con el fin de no afectar a los seres vivos.

En este trabajo de investigación se resolvió el problema: *predecir la posición de las fuentes de reproducción de mosquitos*. En términos generales, esto representó en desarrollar un dispositivo electrónico que es capaz de medir variables climatológicas como la temperatura y humedad, además de la atracción y captura de mosquitos que se encuentran dentro de una región, esto, se sabe de la existencia de mosquitos debido a las condiciones de la región (un jardín). Por otro lado, se resolvió de manera numérica la ecuación de reacción-difusión en una dimensión, aunque esto no nos permite simular el comportamiento de los mosquitos ya que es un término lineal y por lo menos se necesita resolver la ecuación en dos dimensiones, por tanto esto queda pendiente como trabajo a futuro para resolver dicha ecuación agregando una dimensión más para el método de diferencias finitas desarrollado, a su vez elegir valores para los coeficientes de reacción y difusión bajo las condiciones de los parámetros de temperatura y humedad. Esta tesis de maestría representa la continuación del trabajo realizado en la tesis de licenciatura.

Previo a la construcción del dispositivo se trabajó con la configuración de los módulos para la medición de datos como la temperatura y humedad, la comunicación serial a través de Bluetooth y el manejo correspondiente del Arduino Due. La construcción del dispositivo electrónico se resumió en tres etapas. La etapa uno se baso en determinar que método de atracción es más conveniente utilizar ya sea por atracción mediante CO₂ o luz ultravioleta el cual al final de determinó utilizar la luz ultravioleta por su características y mantenimiento. En la etapa dos se trabajó en la fabricación de la base del dispositivo el cual se construyó con un material conocido como MDF. En la etapa tres, se colocaron todos los módulos como el sensor DHT22 y el Bluetooth, además del Arduino Due. Más específico, se puso a trabajar el dispositivo electrónico durante dos horas al día por 131 días el cual comenzó desde el 17 de Marzo de 2020 hasta el 26 de Julio de 2020, para medir

datos respecto a las variables climatológicas y la cantidad de mosquitos. Los mosquitos se contaron de manera manual ya que se había propuesto contarlos de manera automática a través del procesamiento de imagen, sin embargo, se determinó que esto no era muy factible utilizarlo ya que el costo del dispositivo electrónico aumentaba, además de que el dispositivo estaba vulnerable en el ambiente de la región a estudiar. Este dispositivo se puso en un jardín amplio con bastante vegetación ya que es un lugar que debido a dichas condiciones, los mosquitos se pueden reproducir. El dispositivo electrónico cumple con su propósito ya que atrae y captura mosquitos, sin embargo, este dispositivo se puede mejorar en cuestión de tratar de hacerlo más autónomo.

Para simular el comportamiento de los mosquitos, primeramente se aprendieron diferentes temas respecto a resolver problemas de valores iniciales con los métodos de Euler, Runge Kutta y diferencias finitas tanto para ecuaciones parabólicas e hiperbólicas. Los métodos de Euler y Runge Kutta se aprendieron para poder comprender como se aproxima la solución exacta de manera numérica, así mismo el entendimiento del método de diferencias finitas. En este caso la ecuación de reacción-difusión es una ecuación parcial del tipo parabólico la cual fue resuelta con el método de diferencias finitas con coeficientes de difusión y de reacción constantes. Parte del trabajo a futuro es poder resolver esta ecuación diferencial parcial parabólica mediante el método de líneas que prácticamente es resolver la parte temporal con el método de Runge Kutta de cuarto orden y con diferencias finitas la parte espacial.

Palabras clave: Mosquitos, dispositivo electrónico, módulos de medición, DTH22, ecuación de reacción difusión, solución numérica.

Abstract

The mosquito has been considered the most lethal animal in the world due to the number of deaths caused by the transmission of diseases through its bite to living beings and that most of them do not have a vaccine as such, since if the infected living being is not treated in time, it tends to die. Knowing the position of the sources of reproduction of mosquitoes allows us to carry out work to fumigate said area and not contaminate other areas where it is not required, in order not to affect living beings.

In this research work the problem was solved: *To predic the positions of mosquito breeding sources*. In general terms, this represented in developing an electronic device that is capable of measuring climatological variables such as temperature and humidity, in addition to the attraction and capture of mosquitoes that are within a region, this is known of the existence of mosquitoes due to to the conditions of the region (a garden). On the other hand, the reaction-diffusion equation was solved numerically in one dimension, although this does not allow us to simulate the behavior of mosquitoes since it is a linear term and at least it is necessary to solve the equation in two dimensions, for both this remains pending as future work to solve said equation adding one more dimension for the finite difference method developed, in turn choosing values for the reaction and diffusion coefficients under the conditions of the temperature and humidity parameters. This master's thesis represents the continuation of the work done in the bachelor's thesis.

Prior to the construction of the device, we worked with the configuration of the modules for the measurement of data such as temperature and humidity, serial communication through Bluetooth and the corresponding handling of the Arduino Due. The construction of the electronic device was summarized in three stages. Stage one was based on determining which method of attraction is more convenient to use either by attraction by means of CO₂ or ultraviolet light, which at the end was determined to use ultraviolet light due to its characteristics and maintenance. In stage two, we worked on the manufacture of the base of the device which was built with a material known as MDF. In stage three, all the modules like the DHT22 sensor and the Bluetooth were placed, in addition to the Arduino Due. More specifically, the electronic device was put to work for two hours a day for 131 days, which began from March 17, 2020 to July 26, 2020, to measure data regarding weather variables and the number of mosquitoes. Mosquitoes were counted manually since it had been proposed to count them automatically through image processing, however, it

was determined that this was not very feasible to use since the cost of the electronic device increased, in addition to the fact that the device was vulnerable in the environment of the region to study. This device was placed in a large garden with a lot of vegetation since it is a place where due to these conditions, mosquitoes can reproduce. The electronic device serves its purpose as it attracts and captures mosquitoes, however, this device can be improved in a matter of trying to make it more autonomous.

To simulate the behavior of mosquitoes, different topics were first learned about solving initial value problems with the methods of Euler, Runge Kutta and finite differences for both parabolic and hyperbolic equations. The methods of Euler and Runge Kutta were learned to understand how the exact solution approximates in a numerical way, as well as the understanding of the finite difference method. In this case, the reaction-diffusion equation is a partial equation of the parabolic type which was solved with the finite difference method with constant diffusion and reaction coefficients. Part of the future work is to be able to solve this parabolic partial differential equation through the method of lines, which is practically solving the temporal part with the fourth order Runge Kutta method and with finite differences the spatial part.

Capítulo 1

Introducción

En la actualidad, se sabe que existe una gran variedad de especies de insectos. Estos se encuentran en muchos lados, ya sea el tipo de hábitad en la que nos encontremos. En particular, un problema que se presenta a esto son las plagas y generalmente se encuentran en los huertos donde se practica la agricultura de frutos o vegetales. Dependiendo de que es lo que se cultive, esto representa un cierto tipo de plaga.

Otro problema que se presenta por los insectos, son aquellos que transmiten enfermedades. En particular, hablamos de los mosquitos, ya que es un tema que últimamente ha estado en boca de muchos, pues las enfermedades que transmiten a través de su picadura han hecho que muchos seres vivos sean infectados y en muchos de los casos tienden a morir.

Generalmente, cuando se piensa en animales peligrosos, lo primero que nos viene a la mente son los tiburones si es que nos encontramos en el mar, en serpientes si sabemos que nos encontramos en una selva, por mencionar algunos. Sin embargo, el animal más letal del mundo es el mosquito, responsable de la muerte de más de 725,000 personas cada año, frente a las 50,000 muertes provocadas por las serpientes o las apenas 10 personas que matan los tiburones. Los seres humanos, por cierto, se sitúan en segundo lugar en este listado, justo después del mosquito, y son responsables de 475,000 personas muertas al año [1].

Existe una gran familia de mosquitos tales como: Los culícidos (Culicidae) son una familia de dípteros nematóceros conocidos coloquialmente como mosquitos, y en algunas partes de América como zancudos. Incluye, entre otros, los géneros Anopheles, Culex, Psorophora, Ochlerotatus, Aedes, Sabethes, Culiseta y Haemagogus. En la actualidad existen un total de 39 géneros y 135 subgéneros reconocidos con algo más de 2,500 especies. El descubrimiento de nuevas especies así como cambios en la sistemática y las dificultades en la aceptación de algunos taxones hace imposible reflejar cifras exactas. Son insectos voladores, que poseen un cuerpo delgado y patas alargadas; el tamaño de los

adultos varía según las especies, pero rara vez superan los 15 mm. Las larvas y pupas se desarrollan en el agua.

Este pequeño insecto, cuenta con una capacidad para transmitir virus y otros parásitos que causan enfermedades devastadoras es, precisamente, lo que lo hace tan peligroso. Así, por ejemplo, la malaria, que es transmitida por el mosquito *Anopheles*, mata cada año a 400,000 personas (principalmente niños) e incapacita por varios días a más de 200 millones [2]. Otras enfermedades transmitidas por el mosquito incluyen el dengue, que causa entre 50 y 100 millones de casos cada año [3], la fiebre amarilla, que tiene una alta tasa de mortalidad [4], o la encefalitis japonesa, que causa más de 10,000 muertes anuales [5], sobre todo en Asia. Sin olvidar, el virus del Zika [6], que tiene efectos neurológicos graves y de largo plazo en bebés de madres infectadas durante el embarazo.

La existencia de insectos, en este caso los mosquitos son de suma importancia en la actualidad. Los mosquitos son conocidos por transmitir enfermedades al momento de realizar una picadura. El virus más reciente transmitido por los mosquitos es conocido como el **chikungunya**, en la actualidad, este virus no tiene una vacuna como tal, así que debe de ser tratada de forma adecuada hacia el paciente infectado, de lo contrario este podría morir. Otra enfermedad famosa de este insecto es el virus conocido como **zika** que en particular es transmitido por mosquitos de la familia *Aedes*. Una vez infectado se presentan los síntomas, generalmente leves y de 2 a 7 días de duración, consisten en fiebre, erupciones cutáneas, conjuntivitis, dolores musculares y articulares, malestar y cefaleas. La mayoría de las personas infectadas son asintomáticas. Así como el virus de la chikungunya, este no tiene una vacuna como tal pero es afortunadamente es tratable. Otro vector de enfermedades de transmisión infecciosas por mosquitos son la fiebre amarilla, la malaria y dengue.

Existen más de 2500 especies de mosquitos y estos animales se encuentran en todas las regiones del planeta a excepción de la Antártida. De hecho, los mosquitos son muy buenos para adaptarse a nuevos ambientes y a cualquier intervención que usemos contra ellos. Por ejemplo, *Aedes aegypti* (el vector de la fiebre amarilla, zika y dengue, entre otros) se ha adaptado perfectamente al ambiente urbano: se alimenta exclusivamente de sangre humana y pone sus huevos en cualquier tipo de recipiente fuera o dentro de las casas. Muchas especies de mosquito, incluyendo el *Anopheles*, han desarrollado resistencia contra una variedad de insecticidas comúnmente usados y han cambiado su comportamiento (se alimentan fuera de la casa y más temprano) para evitar las redes mosquiteras y las casas fumigadas con insecticida.

Como señala el entomólogo del Instituto de Salud Global de Barcelona (ISGlobal), Krijn Paaijmans, “Los mosquitos son criaturas difíciles. Están constantemente evitando cualquier cosa que hagamos contra ellos”. Paaijmans es parte de un gran equipo que trabaja en MALTEM, un proyecto financiado por la Obra Social “la Caixa” y la Fundación Bill y Melinda Gates, cuyo objetivo es detener la transmisión de la malaria en el distrito de Magude, en el sur de Mozambique, en el 2020. “Si queremos eliminar la malaria, necesitamos mejor vigilancia y mejores datos para desarrollar intervenciones más rentables.

Esto implica formular las preguntas correctas y utilizar las herramientas adecuadas". Se pone como ejemplo utilizar el mismo tipo de trampas fuera y dentro de las casas para medir correctamente la transmisión residual de malaria (la transmisión, que ocurre en el exterior, donde las redes mosquiteras y la fumigación no son eficaces) [7][8].

ISGlobal también está trabajando en nuevas herramientas contra las cuales el mosquito tiene más difícil desarrollar resistencia. Por ejemplo, el equipo de Paaijmans está desarrollando barreras electromagnéticas que repelen a los mosquitos, mientras que el equipo de Carlos Chaccour, también de ISGlobal, está centrado en la ivermectina, un fármaco anti-parasitario que mata a los mosquitos que se alimentan de animales o individuos tratados.

Conociendo aún más como está la situación del comportamiento de los mosquitos y las afectaciones que causan entre los seres vivos, esto implica que debemos tener cuidado al como se debería de tratarlos, por esto, debido a que los mosquitos son capaces de adaptarse a ciertas condiciones, Para resolver este problema es conveniente el uso de herramientas tecnológicas y modelos teóricos eficientes que nos permitan amenizar las transmisión de enfermedades de estos insectos. En las próximas secciones hablaremos de componentes electrónicos y modelos teóricos para poder resolver este problema.

1.1. Componentes electrónicos

Arduino es una empresa de hardware y software de código abierto, comunidad de usuarios y proyectos que diseña y fabrica microcontroladores de una sola placa y kits de microcontroladores para construir dispositivos digitales. Sus productos están licenciados bajo la Licencia Pública General Menor de GNU (LGPL) o la Licencia Pública General de GNU (GPL), lo que permite la fabricación de tableros Arduino y la distribución de software por cualquier persona. Las placas Arduino están disponibles comercialmente en forma preensamblada o como kits de bricolaje.

Los diseños de placas Arduino utilizan una variedad de microprocesadores y controladores. Las tarjetas están equipadas con conjuntos de pines de entrada / salida (E / S) digitales y analógicas que pueden conectarse a varias tarjetas de expansión o placas de prueba (para prototipos) y otros circuitos. Las placas cuentan con interfaces de comunicaciones en serie, incluido el bus serie universal (USB) en algunos modelos, que también se utilizan para cargar programas desde computadoras personales. Los microcontroladores se pueden programar utilizando lenguajes de programación C y C ++. Además de usar cadenas de herramientas de compilación tradicionales, el proyecto Arduino proporciona un entorno de desarrollo integrado (IDE) basado en el proyecto de lenguaje de procesamiento.

Para este proyecto nos enfocaremos en particular en la tarjeta Arduino Due.

Especificaciones técnicas	
Microcontrolador	AT91SAM3X8E
Tensión de funcionamiento	3.3V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-16V
Pines de E / S digitales	54 (de los cuales 12 proporcionan salida PWM)
Pines de entrada analógica	12
Pines de salida analógica	2 (DAC)
Corriente de salida de CC total en todas las líneas de E / S	130 mA
Corriente DC para Pin de 3.3V	800 mA
Corriente DC para Pin 5V	800 mA
Memoria flash	512 KB, todos disponibles para las aplicaciones del usuario
SRAM	96 KB (dos bancos: 64 KB y 32 KB)
Velocidad de reloj	84 MHz
Longitud	101,52 mm
Anchura	53,3 mm
Peso	36 g

Tabla 1.1: Especificaciones técnicas del Arduino Due. [9]

1.1.1. Arduino Due

En el mercado existen múltiples tarjetas que pueden ser programables con el fin de realizar proyectos que van desde los muy simples hasta los muy complejos. Cada tarjeta ofrece diferentes ventajas, hay tarjetas que son micro-controladores y otras que son mini-computadoras, una de las principales diferencias entre ellas es la cantidad de energía de consumo. En lo que concierne al dispositivo electrónico, uno de los objetivos de este proyecto es poder optimizar el consumo de energía lo más posible, por lo que es vital que el dispositivo electrónico pueda durar el mayor tiempo. Para esto se decidió utilizar un micro-controlador conocido como Arduino ya que los costos de éste son baratos y además hay una gran cantidad de documentación acerca de este dispositivo programable.

Arduino tiene una línea de micro-controladores diseñados para distintos proyectos, en sí, existe un producto conocido como Arduino Due que tiene bastantes características tales como se muestran en la tabla 1.1.

En resumen, como podemos ver que Arduino Due es una placa de micro-controlador basada en la CPU Atmel SAM3X8E ARM Cortex-M3. Es la primera placa Arduino basada

en un microcontrolador de núcleo ARM de 32 bits. Tiene 54 pines de entrada / salida digital (de los cuales 12 se pueden usar como salidas Pulse Width Modulation PWM), 12 entradas analógicas, 4 UART (puertos serie de hardware), un reloj de 84 MHz, una conexión compatible con USB OTG, 2 DAC (digital a analógico) , 2 TWI, un conector de alimentación, un encabezado SPI, un encabezado JTAG, un botón de reinicio y un botón de borrado [10].

Es importante recalcar que el funcionamiento de la tarjeta Arduino Due es de 3.3 V. El voltaje máximo que los pines de E / S pueden tolerar es 3.3 V. La aplicación de voltajes superiores a 3.3 V a cualquier pin de E / S podría dañar la placa.

El Arduino Due se puede alimentar a través del conector USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente. La alimentación externa (no USB) puede provenir de un adaptador de CA a CC o de una batería. El adaptador se puede conectar enchufando un conector de centro positivo de 2.1 mm en el conector de alimentación de la placa. Los cables de una batería se pueden insertar en los encabezados de clavija Gnd y Vin del conector POWER.

La placa puede funcionar con una fuente externa de 6 a 20 voltios. Sin embargo, si se suministra con menos de 7 V, el pin de 5 V puede suministrar menos de cinco voltios y la placa puede ser inestable. Si usa más de 12V, el regulador de voltaje puede sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 voltios.

El SAM3X tiene 512 KB (2 bloques de 256 KB) de memoria flash para almacenar código. El gestor de arranque se graba previamente en fábrica de Atmel y se almacena en una memoria ROM dedicada. La SRAM disponible es de 96 KB en dos bancos contiguos de 64 KB y 32 KB. Se puede acceder a toda la memoria disponible (Flash, RAM y ROM) directamente como un espacio de direccionamiento plano. El Due se puede programar con el software Arduino Arduino (IDE).

En la Fig. 1.1 se muestra el tipo de conexión del dispositivo donde cualquiera de los puertos USB se puede usar para programar la placa. Aquí, *Native port* (puerto nativo) es utilizado una vez que la tarjeta ya ha sido programada y *Programming port* (puerto de programación) se utiliza cuando se esta programando la tarjeta Arduino, aunque se recomienda usar el puerto de programación debido a la forma en que se maneja el borrado del chip [11].

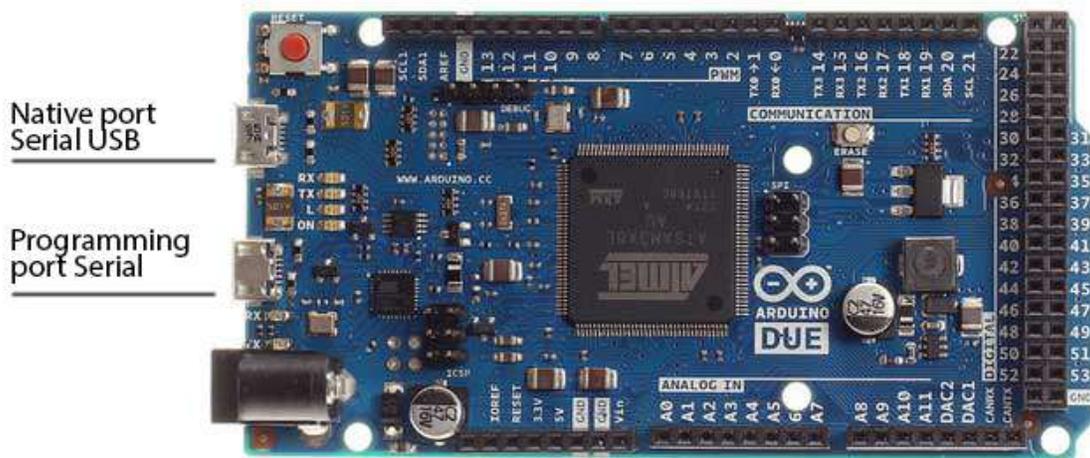


Figura 1.1: Arduino Due mostrando los dos tipos de conexión para ser programada, tanto como native port y programming port.

1.1.2. Módulo DHT22

Uno de los sensores de medición de datos para la temperatura y humedad más popular en la actualidad es el sensor de medición DHT22 (también llamado como AM2302). Este sensor cuenta con una salida digital señal calibrada. Utiliza una técnica exclusiva de recolección de señal digital y tecnología de detección de humedad, asegurando su estabilidad. Sus elementos de detección están conectados con una computadora de un solo chip de 8 bits. Cada sensor de este modelo tiene compensación de temperatura y está calibrado en una cámara de precisión y el coeficiente de calibración se guarda en el tipo de programa en la memoria OTP, cuando el sensor está detectando, citará el coeficiente de la memoria. Este módulo es pequeño y de bajo consumo, contiene 4 pines, lo que hace que la conexión sea muy conveniente [12].

En la tabla 1.2 se muestran las especificaciones técnicas del módulo DHT22 y en la Fig. 1.2 se muestra la forma del sensor, además de indicar los correspondientes pines.

El módulo DHT22, para que pueda medir la temperatura y humedad contiene dos sensores, es decir, consisten en un componente de detección de humedad, un sensor de temperatura NTC (o termistor) y un IC en la parte posterior del sensor.

Para medir la humedad utilizan el componente sensor de humedad que tiene dos electrodos con sustrato de retención de humedad entre ellos. Entonces, a medida que cambia la humedad, cambia la conductividad del sustrato o cambia la resistencia entre estos electrodos. Este cambio en la resistencia es medido y procesado por el IC que lo prepara para ser leído por un microcontrolador. Por otro lado, medir la temperatura estos sensores usan un sensor de temperatura NTC o un termistor.

Modelo	DHT22
Fuente de alimentación	3.3-6V DC
Señal de salida	Señal digital a través de un solo bus
Elemento de sensor	Condensador de polímero
Rango de operación	humedad 0-100 % HR; temperatura -40 80 Celsius
Exactitud	humedad ± 2 % RH (Max ± 5 % RH); temperatura ± 0.5 Celsius
Resolución o sensibilidad	humedad 0.1 % HR; temperatura 0.1 Celsius
Repetibilidad	humedad ± 1 % HR; temperatura ± 0.2 Celsius
Humedad	± 0.3 % HR
Estabilidad a largo plazo	± 0.5 % HR / año
Periodo de detección	Promedio: 2s
Intercambiabilidad	totalmente intercambiable
Dimensiones	Tamaño grande 22 * 28 * 5 mm

Tabla 1.2: Especificaciones técnicas del módulo DHT22.

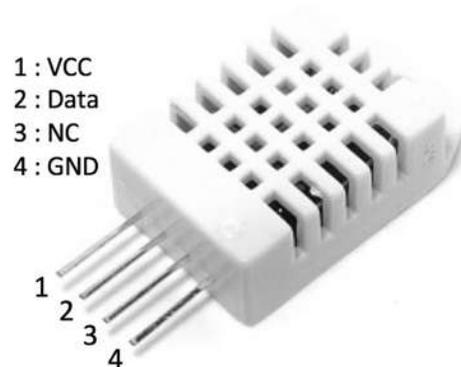


Figura 1.2: Sensor DHT22 donde se muestra la correspondencia de cada pin.

Un termistor es en realidad una resistencia variable que cambia su resistencia con el cambio de temperatura. Estos sensores se fabrican mediante la sinterización de materiales semiconductores como la cerámica o los polímeros para proporcionar cambios más grandes en la resistencia con sólo pequeños cambios en la temperatura. El término “NTC” significa “Coeficiente de temperatura negativo”, lo que significa que la resistencia disminuye con el aumento de la temperatura [13].

1.1.3. Módulo Bluetooth HC-06

El módulo Bluetooth HC-06, está diseñado para transmitir datos a distancias cortas ya sea por teléfono celular o a través de una computadora. Con este tipo de conexión no nada más se pueden transmitir datos, si no también es posible hacer modificación al código, hacer análisis de componentes electrónicos, entre otras cosas, todo desde un dispositivo externo. Esto es una buena ventaja, ya que no es necesario acceder al Arduino Due de manera física.

El Módulo Bluetooth HC-06 permite la configuración de algunos de sus parámetros de funcionamiento mediante el uso de comando *AT*. Los comandos *AT* son una lista de comandos que inician siempre con las letras *AT*, estos comandos son enviados por medio de un puerto Serie por lo que necesitaremos un Arduino o Conversor USB Serial para poder enviar los comandos desde nuestra PC o cualquier otro dispositivo electrónico.

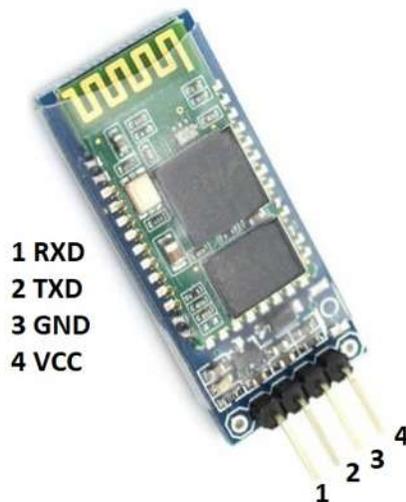


Figura 1.3: Módulo bluetooth HC-06.

Algunos parámetros que vienen por defecto se muestran a continuación:

- Nombre por defecto: "linvor" o "HC-06".
- Código de emparejamiento por defecto: 1234.
- La velocidad por defecto (baud rate): 9600.

El Módulo Bluetooth HC-06 tiene dos estados de funcionamiento los cuales es importante conocer:

- Modo *At* (Desconectado).

- Entra a este modo tan pronto alimentamos el módulo, y cuando no se ha establecido una conexión bluetooth con ningún otro dispositivo.
 - El LED del módulo está parpadeando (frecuencia de parpadeo del LED es de 102 ms).
 - En este modo es cuando se debe enviar los comandos AT en caso se quiera configurar algún parámetro, si se envían otros datos diferentes a los comandos AT, el HC-06 los ignorará.
- Modo Conectado.
- Entra a este modo cuando se establece una conexión con otro dispositivo bluetooth.
 - El LED permanece encendido sin parpadear.
 - Todos los datos que se ingresen al HC-06 por el Pin RX se transmiten por bluetooth al dispositivo conectado, y los datos recibidos se devuelven por el pin TX. La comunicación es transparente para el programador.
 - En este Modo el HC-06 no puede interpretar los comandos AT.

El Módulo HC-06 viene configurado de fábrica como “Esclavo”(Slave) y no puede ser cambiado a “Maestro” como si lo permite el módulo HC-05.

El funcionamiento del HC-06 cuenta con una comunicación en serie. El módulo Bluetooth en el otro extremo recibe los datos y los envía a Arduino Due a través del pin TX del módulo Bluetooth (pin RX de Arduino). El código alimentado a Arduino verifica los datos recibidos y los compara. Si los datos recibidos son 1, el LED se enciende y se apaga cuando los datos recibidos son 0 [14].

1.2. Modelos matemáticos

Los sistemas de reacción-difusión son modelos matemáticos que describen cómo una o más sustancias distribuidas en el espacio cambian bajo la influencia de dos procesos: reacciones químicas locales en las que las sustancias se transforman las unas en las otras, y difusión, que provoca que las sustancias se expandan en el espacio. El resultado de este proceso es una configuración estable en la que la composición química es no uniforme en un dominio espacial.

En los últimos tiempos, los sistemas de reacción-difusión han atraído mucho interés como modelo prototipo para la formación de patrones [15]. Los patrones antes mencionados (frentes, espirales, objetivos, hexágonos, rayas y solitones disipativos) se pueden encontrar en varios tipos de sistemas de reacción-difusión a pesar de las grandes discrepancias, en los términos de reacción local. También se ha argumentado que los procesos

de reacción-difusión son una base esencial para los procesos relacionados con la morfogénesis en biología [16] presentes en los patrones de las pieles de los animales y la pigmentación de la piel [17][18]. Otras aplicaciones de las ecuaciones de reacción-difusión incluyen invasiones ecológicas [19], propagación de epidemias [20], crecimiento tumoral [21][22][23] y curación de heridas [24]. Otra razón para el interés en los sistemas de reacción-difusión es que, aunque son ecuaciones diferenciales parciales no lineales, a menudo existen posibilidades para un tratamiento numérico.

Los sistemas de reacción-difusión se aplican a la modelización de procesos tanto químicos como dinámicos de naturaleza no química. Encontramos ejemplos de tales aplicaciones en biología, geología, física y ecología. Matemáticamente, los sistemas de reacción-difusión tienen la forma de ecuaciones parabólicas en derivadas parciales y pueden representarse bajo la forma general:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + R(u), \quad (1.1)$$

donde $u(x, t)$ representa una función vectorial desconocida, D representa el coeficiente de difusión y R el coeficiente de reacción. Estos coeficientes están relacionados entre sí, es decir, basta con cambiarlos para estudiar diferentes sistemas de comportamiento.

Las soluciones de las ecuaciones de reacción-difusión muestran un amplio rango de comportamientos, incluyendo la formación de ondas viajeras y fenómenos de tipo onda, así como otros patrones auto-organizados como bandas, hexágonos o estructuras más complicadas como solitones disipativos.

Desde que en 1952 Alan Turing los propuso como "la base química de la morfogénesis", los sistemas de reacción-difusión se han utilizado para modelizar diversos procesos biológicos de formación de patrones.

En los sistemas químicos, las reacciones químicas son la fuente de los componentes moleculares y la difusión es el proceso físico clásico basado en el movimiento browniano. Sin embargo, en el caso del desarrollo de tejidos, tanto la reacción como la difusión son procesos mucho más complejos: la reacción implica la producción y el consumo de moléculas por parte de las células; el transporte a través de tejidos, aunque capaz de generar gradientes moleculares, es un proceso mucho más complejo que la simple difusión. De ahí que se haya propuesto el término "reactor-difusión" para la aplicación del mecanismo de Turing a procesos biológicos [25].

1.3. Inteligencia artificial

La capacidad de aprender se considera uno de los atributos distintivos del ser humano y de los animales, esto ha sido una de las principales áreas de investigación de la Inteligencia Artificial desde sus inicios.

En los últimos años se ha visto un crecimiento acelerado en la capacidad de generación y almacenamiento de información, debido a la creciente automatización de procesos y a los avances desarrollados en cualquier área que requiera del uso de una computadora. Esto ha llevado a desarrollar una gran cantidad de herramientas y técnicas que tienen que ver con el análisis de información. En este aspecto, el desarrollo en el área de aprendizaje ha sido fundamental. El área de aprendizaje en general trata de construir programas que mejoren su desempeño automáticamente con la experiencia.

Desde el comienzo de las computadoras se cuestionó si serían capaces de aprender. El darles la capacidad de aprendizaje a las máquinas abre una amplia gama de nuevas aplicaciones. El entender también como pueden aprender las máquinas nos puede ayudar a entender las capacidades y limitaciones humanas de aprendizaje, siempre y cuando se desarrollen más mecanismos en función al comportamiento humano, como las prótesis robóticas [26] que pueden ser controlados por los humanos.

El aprendizaje humano en general es muy diverso e incluye, la adquisición de conocimiento, desarrollo de habilidades a través de instrucción y práctica, organización de conocimiento, descubrimiento de hechos, entre otros.

De la misma forma el aprendizaje computacional o aprendizaje máquina (conocido en inglés como Machine Learning) se encarga de estudiar y modelar computacionalmente los procesos de aprendizaje en sus diversas manifestaciones.

En general, se busca construir programas que mejoren automáticamente con la experiencia, es decir, se registra cada proceso ocurrido al que se les dan ciertas prioridades para que así los procesos sean mejorados. Aunque no se tienen programas que aprendan tan bien como los humanos, actualmente existen algoritmos que han probado ser muy efectivos para ciertas tareas tales como las industrias de coches, la bolsa, física computacional, entre otros.

Actualmente existe una gran cantidad de métodos de aprendizaje tales como: árboles de decisión con el uso de reglas de clasificación, métodos de clasificación, redes y regresiones no-lineales, modelos gráficos de dependencias probabilísticas y muchos otros más.

Algunos de los ejemplos que usan los algoritmos de aprendizaje son sistemas de reconocimiento de voz (utilizado por los bancos), manejo de vehículos autónomos (utilizado por los carros Tesla). Por otro lado, existe una gran cantidad de aplicaciones reales relacionadas con descubrimiento de conocimiento en base de datos que utilizan algún algoritmo

de aprendizaje. Estas aplicaciones se encuentran en áreas tales como:

- Astronomía: clustering y clasificación de cuerpos celestes.
- Biología molecular: predicción de sustancias cancerígenas, genoma humano.
- Aspectos climatológicos: predicción de tormentas.
- Medicina: caracterización y predicción de enfermedades.
- Física: análisis de ondas gravitacionales.
- Mecatrónica: aprendizaje de tareas en robótica.

y éstas son algunas de las áreas en las que se puede aplicar el aprendizaje computacional.

En este trabajo se estudia el algoritmo de aprendizaje perteneciente al tipo de Métodos de clasificación y regresiones no-lineales. Este método es conocido como **Redes Neuronales Artificiales** (RNAs) e implementa un algoritmo de entrenamiento llamado **retropropagación** (en inglés es conocido como **Backpropagation algorithm**).

Aunque muchos tipos de algoritmos han sido propuestos en los últimos años, sin embargo el algoritmo de retropropagación es el más destacado en cuestión de clasificación. En las siguientes secciones se describirá a detalle el funcionamiento de dicho método y al final se darán algunos ejemplos con la finalidad de comprender el funcionamiento de las Redes Neuronales Artificiales.

En los siguientes dos capítulos se describe la forma en la que se configuran todos estos componentes tanto teóricos como electrónicos para hacer uso en nuestro problema.

1.4. Trabajo preliminar

En este trabajo de tesis, el objetivo principal es predecir las posiciones de las fuentes de reproducción de mosquitos. Previamente a esto, es importante analizar ciertas cosas como la región en donde se va a estudiar los mosquitos, sus variables climatológicas, la forma en la que se desea atraer y atrapar a dichos mosquitos. De acuerdo a artículos publicados por investigadores biólogos los mosquitos suelen reproducirse en lugares donde hay agua como lagos, ríos, estanques, incluso basta con 1 cm^3 de agua para poder reproducirse, con esta información, la región a estudiar puede ser en la zona donde vivimos y para esta ocasión se utilizará la zona de la Universidad Michoacana de San Nicolás de Hidalgo. Para el estudio de la variables climatológicas se medirán ciertos parámetros tales como la temperatura, humedad y precipitación. Finalmente para atraer a los mosquitos se utilizará

una técnica de atracción de luz ultravioleta y para atrapar a los mosquitos se utilizará un ventilador en dirección contraria a la entrada de mosquitos.

En esta tesis de maestría se continuó con el trabajo que faltó por terminar en la tesis de licenciatura [27] donde se atacó el problema referido a encontrar las posiciones de las fuentes de reproducción de mosquitos. En términos generales, el trabajo se dividió en tres partes: métodos numéricos, inteligencia artificial y dispositivos electrónicos. Para los métodos numéricos, el grupo de trabajo resolvió de manera numérica la ecuación de Reacción-Difusión donde yo tomé dichas simulaciones y las interprete como la propagación de los mosquitos dentro de una región. Para la inteligencia artificial se programó una Red Neuronal Artificial la cual fue utilizada para clasificar las simulaciones numéricas. Finalmente, para el dispositivo electrónico se construyó un primer prototipo que cuenta mosquitos y mide los datos de las variables climatológicas. Este prototipo fue diseñado con una herramienta digital de modelado 3D, donde solo se construyó con madera la parte estructural del dispositivo, por el lado de la medición de datos, no se configuró ningún módulo.

Inicialmente en este proyecto se necesita recolectar información acerca de los mosquitos y el ambiente que los rodea, para ello es necesario conocer cierta información como la cantidad de mosquitos y las variables climatológicas en particular la temperatura, humedad y precipitación [28]. Para obtener estos datos es necesario contar con un dispositivo electrónico que contiene sensores que permiten medir dichos datos, además de un sistema de captura de mosquitos y conteo. En los siguientes capítulos se describen las herramientas que se utilizaron y que es lo que necesita saber en un principio para poder configurarlos de manera adecuada.

En el capítulo 2, se hablará acerca de los componentes electrónicos que se utilizaron para la construcción del dispositivo electrónico. En el capítulo 3, se describen los métodos numéricos que se utilizaron para resolver la ecuación que permitirá simular el comportamiento de los mosquitos. En el capítulo 4, tratamos el problema a resolver, es decir, se describe la forma en la que construyó el dispositivo electrónico además los resultados de la solución numérica de la ecuación reacción-difusión. Finalmente, en el capítulo 5, se habla de las conclusiones y discusiones del tema a trabajar.

Capítulo 2

Elementos electrónicos

En este capítulo veremos la programación de los componentes electrónicos y la construcción de las trampas. Conjuntamente, se explicarán las razones por las cuales se decidió construir y/o programar de esta forma.

2.1. Arduino Due

Como hemos explicado anteriormente el Arduino Due ofrece una gran cantidad de ventajas, básicamente la velocidad de reloj y el consumo de energía como se especifica en la tabla 1.1. Adicionalmente a la tarjeta, ésta contiene una configuración de pines a la que se le tiene que tener cuidado, en otras palabras, de todos los pines de la tarjeta no todos se pueden utilizar con total libertad, ya que hay pines que se manejan a cierto voltaje el cual no debe de exceder los 3.3 V, pines que no se pueden utilizar por ningún motivo ya que se puede provocar un corto circuito y dañar tanto la placa como el circuito conectado a ella, también hay pines que tienen más potencia que otros. Todo esto es debido a que el fabricante así decidió hacer dicho diseño y aunque existe un documento donde se especifican estos diseños, sin embargo, no se estudiarán en este trabajo. Algo que si se debe saber de los pines es como esta su distribución, en la Fig. 2.1 y Fig. 2.2 se muestra dicha distribución.

Como podemos observar en las tablas de la Fig. 2.1 y 2.2, en cada puerto se muestran 3 columnas donde la primera columna significa un identificador de comunicación¹, la segunda columna nos indica cual es el puerto físico en la placa Arduino Due y la tercera columna indican comentarios respecto a los pines. Además de la representación en código binario y en código hexadecimal donde estos códigos simbolizan a la disponibilidad de

¹El identificador se utiliza en Atmel Studio que es un programador del micro-controlador de la tarjeta Arduino Due, el cual se describe más a fondo en la sección 2.2.

Puerto A	Pin fisico en Arduino DUE	comentarios	
PA0	CANTX		1
PA1	CANRX		1
PA2	A7		1
PA3	A6		1
PA4	A5		1
PA5	EEXTINT		0
PA6	A4		1
PA7	PIN31		1
PA8	RX0	Fuero "Programming atmega16u2"	0
PA9	TX0		0
PA10	RX2		1
PA11	TX2		1
PA12	RX1		1
PA13	TX1		1
PA14	PIN23		1
PA15	PIN24		1
PA16	A0		1
PA17	SDA1		1
PA18	SCL1		1
PA19	PIN42		1
PA20	PIN43		1
PA21	TXL	LED TX	0
PA22	A3		1
PA23	A2		1
PA24	A1		1
PA25	MISO	conector denominado "SPI"	0
PA26	MOSI		0
PA27	SPCK		0
PA28	PWM10		1
PA29	PWM4		1
			0
			0
23 pines			

	no disponible/ocupado
	disponible
	precaucion
	advertencia

Puerto B	Pin fisico en Arduino DUE	comentarios	
PB0	ETX_CLK		0
PB1	ETX_EN		0
PB2	ETXD0		0
PB3	ETXD1		0
PB4	ERX_DV		0
PB5	ERXD0		0
PB6	ERXD1		0
PB7	ERX_ER		0
PB8	EMDC		0
PB9	EMDIO		0
PB10	OUTGCBOF		0
PB11	OUTGID		0
PB12	SDA	resistencia pull up de 1.5k	1
PB13	SCL		1
PB14	PIN53		1
PB15	DAC0		1
PB16	DAC1		1
PB17	A8		1
PB18	A9		1
PB19	A10		1
PB20	A11		1
PB21	PIN52		1
PB22			0
PB23	SS3		0
PB24			0
PB25	PWM2		1
PB26	PIN22		1
PB27	PWM13	conectado a led	0
PB28	JTAG_TCK	Conector denominado "JTAG"	0
PB29	JTAG_TDI		0
PB30	JTAG_TDO		0
PB31	JTAG-TMS		0
12 pines			

Figura 2.1: Distribución de los puertos A y B, además de un cuadro central donde indica que significa cada color de Arduino Due.

uso de tales pines; por ejemplo, para el puerto A si se quisiera utilizar los primeros 4 pines (PA0 - PA3) la representación en binario es 1111 y en hexadecimal es F, ahora si se quisiera utilizar los siguientes 4 pines (PA4 - PA7) la representación en binario es 1011 y en hexadecimal es D, debido a que el pin PA5 no esta disponible para ser utilizado. Para saber más acerca de las conversiones de decimal, binario y hexadecimal véase el apéndice A.

En la tabla de la Fig. 2.1 se muestran dos puertos en particular el puerto A y el puerto B. En el puerto A tenemos 21 pines marcados con color verde, 4 pines marcados con color rosa, 3 pines marcados con color amarillo y 2 pines marcados con color rojo. El puerto B, contiene 16 pines en color rosa, 7 en color amarillo y 10 pines en color verde. Una característica particular de estos dos puertos es que trabajan a baja potencia porque se recomienda trabajar con sensores de bajo consumo, de lo contrario se puede dañar la placa.

En la tabla de la Fig. 2.2 se muestran a los puertos C y D. En el puerto C, se tienen 23 pines en color verde, 6 en color rosa y 2 en color rojo. Para el puerto D, se tienen 11 puertos en verde, por lo que es el único puerto que contienen todos los puertos disponibles. Estos puertos tienen una característica en particular ya que trabajan con una mayor potencia de corriente.

2. Lenguaje de programación de alto nivel C/C++.
3. Rutinas de montaje fáciles de integrar.
4. Opciones de compilación / enlace fáciles de manipular.
5. Capacidad de depurar.
6. Capacidad para ejecutar código en un simulador (alterar registros, código de tiempo, manipular puertos, etc.).
7. Fácil traducción a la familia ATMega328 independiente y / u otros chips Atmel.
8. Atmel Studio funciona con toda la línea de chips Atmel.
9. Se puede utilizar las librerías de Arduino.

Éstas son algunas de las razones por la cual se utilizará Atmel Studio, principalmente es por la encriptación que se le puede hacer al programa creado ya que en la IDE de Arduino se tiene que pagar por la encriptación. El motivo por el que se quiere encriptar el programa se debe a los derechos del autor.

Para programar Arduino Due en Atmel Studio 7, se comienza por cosas básicas como el encendido y apagado de un LED y utilizar un interruptor para apagar y encender un LED o sistema en general. Con estos dos proyectos básicos se aprenden cosas como la entrada y salida de datos, que básicamente en términos generales es lo que se necesita para programar la mayoría de sensores que se utilizarán en este trabajo de investigación.

Las siguientes líneas representan lo básico del programa.

```
main.c
1  #include <asf.h>
2  int main (void){
3      /* Insert system clock initialization code here (sysclk_init()). */
4      board_init();
5  }
```

Aquí, la primera línea se esta incluyendo un librería predeterminada por Atmel Studio la cual contiene toda la información para poder comunicarse con el chip del Arduino Due. En la siguiente línea contiene una función nombrada como main y que esta función no recibe parámetros como entrada, asimismo, ésta deberá contener unas llaves tipo “{}” donde dentro de ellas estará el contenido de la función. La tercera línea representa la forma de escribir un comentario y por último la cuarta línea se llama una función de la librería *asf.h* la cual permite inicializar la tarjeta en modo programable.

Para hacer cualquier programa en Atmel Studio, es primordial configurar los puertos y pines a utilizar junto con el tipo de señales y velocidades de reloj. Inicialmente, se necesita configurar la señal de reloj para cualquier puerto y dependiendo de cual sea, éste tendrá un registro diferente. Peripheral Manage Control (PMC) optimiza el consumo de energía al controlar todos los relojes periféricos del sistema y del usuario. Como usuario es posible activar la señal de reloj con Peripheral Control Enable Register 0 (PCER0) o desactivar dicha señal con Peripheral Control Disable Register 0 (PCDR0) y en conjunto la instrucción correspondiente para cada puerto es:

- REG_PMC_PCER0 = 0x00000800; – Puerto A.
- REG_PMC_PCER0 = 0x00001000; – Puerto B.
- REG_PMC_PCER0 = 0x00002000; – Puerto C.
- REG_PMC_PCER0 = 0x00004000; – Puerto D.

Como se puede observar, aparece REG_ en la instrucción y esto proviene de la librería *asf.h* que proporciona el mismo Atmel Studio ya que es parte de la comunicación hacia la tarjeta Arduino Due, por tanto siempre que se quiera configurar cualquier cosa de la placa, llevará previamente escrito un REG_. El valor en hexadecimal de cada puerto es proporcionado de acuerdo a los pines a utilizar. Seguidamente a esto, el siguiente paso es configurar el puerto y los correspondientes pines a utilizar. El puerto D tiene 11 pines disponibles que los cuales podemos activarlos todos o algunos basándonos en el código binario, si el valor está en cero entonces el pin esta desactivado, si el valor está en uno entonces está activado. Una vez que se sepa cuales son los pines a utilizar es conveniente convertir el número binario a hexadecimal, tal y como se muestran en los siguientes ejemplos.

Ejemplo 1: Se activan todos los pines del puerto D											
Puerto D	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	PD9	PD10
Pin físico	PIN25	PIN26	PIN27	PIN28	TX3	RX3	PIN29	PMW11	PMW12	PIN30	PIN32
Binario	1	1	1	1	1	1	1	1	1	1	1
Hexadecimal	REG_PIOD_PER = 0x000007FF										

Tabla 2.1: Ejemplo 1: Se activan todos de los pines del puerto D.

Puerto D	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	PD9	PD10
Pin físico	PIN25	PIN26	PIN27	PIN28	TX3	RX3	PIN29	PMW11	PMW12	PIN30	PIN32
Binario	1	0	0	0	1	1	0	1	0	1	1
Hexadecimal	REG_PIOD_PER = 0x000006B1										

Tabla 2.2: Ejemplo 2: Se activan algunos de los pines del puerto D.

Como se puede observar en las tablas 2.1, 2.2 y 2.3, la instrucción REG_PIOD_PER activa los pines elegidos, al observar esta instrucción vemos una parte escrita como PIOD lo

Puerto D	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	PD9	PD10
Pin físico	PIN25	PIN26	PIN27	PIN28	TX3	RX3	PIN29	PMW11	PMW12	PIN30	PIN32
Binario	1	1	1	1	0	0	0	0	0	0	1
Hexadecimal	REG_PIOD_PER = 0x00000F04										

Tabla 2.3: Ejemplo 3: Se activan algunos de los pines del puerto D.

que indica que estamos activando los pines del puerto D, y en general para cualquier otro puerto es PIDx con ($x = A, B, C$ o D). El escribir en código hexadecimal nos da una gran ventaja, ya que a diferencia de la IDE de Arduino, se tendría que escribir una instrucción para cada pin que utilizaría, mientras que en Atmel Studio no, pues se reduce la cantidad de código de manera significativa. Conjuntamente se observa que el código hexadecimal contiene algunos ceros después del 0x, en sí, sirve para indicar a cuantos bits se está trabajando. En este caso es de 8 bits, sin embargo, no es necesario escribir tantos ceros, por ejemplo: $0x00000F04 = 0x00F04 = 0xF04$. Por tanto podemos decir que los ceros que están a la izquierda después del símbolo más significativo está de más.

Ya sea el ejemplo que se elija, para configurarlos como **salida** se necesitan dos instrucciones. La primera instrucción REG_PIOD_PUDR; se refiere a la activación o desactivación de las resistencias pull-up las cuales se encuentran de manera interna dentro de la placa. Siempre que se quieran configurar como salida cualquiera de los pines de cualquier puerto, **siempre se tienen que desactivar las resistencias pull-up**. La segunda instrucción REG_PIOD_OER; se refiere a la configuración como salida de los pines elegidos. Entonces las instrucciones correspondientes son:

- REG_PIOD_PUDR = 0x000007FF;
- REG_PIOD_OER = 0x000007FF;

El siguiente paso es poder activar los pines para que sean manejables, es decir, para el caso de encender un LED necesito mandarle corriente y también necesito limpiar el valor mandado para cuando quiera apagar el LED. Con Set Output Data Register (SODR) le indico al micro-controlador que le envíe corriente y con Clear Output Data Register (CODR) le indico al micro-controlador que limpie todos los registros. Esto se hace con las siguientes dos instrucciones:

- REG_PIOD_SODR = 0x000007FF;
- REG_PIOD_CODR = 0x000007FF;

Para el caso en que el pin se quiera configurar como **entrada**, basta con una sola instrucción. Con Output Disable Register (ODR) se puede poner a los pines deseados como salidas, aquí no es necesario desactivar las resistencias pull-up, de hecho no se debe de

hacer debido a que se está recibiendo corriente lo cual es una manera de proteger a la placa. La instrucción corresponde a

- `REG_PIOD_ODR = 0x000007FF`.

Algo que podemos notar ya sea el tipo de configuración elegida es que para todos los registros, se escribió siempre el mismo valor hexadecimal, esto no es coincidencia ya que cada instrucción esta basada en código binario, como se había comentado si es cero está desactivado y si es uno esta activado, entonces todos los registros los queremos activar, es decir, poner el valor en uno y el código hexadecimal ya no lo está indicando. Por esta razón es que en todos los registros se pone el mismo valor. Como dato final, una instrucción que no se debe de olvidar es programar ciclo que se repita una gran cantidad de veces, ya que al ser un micro-controlador lo que se desea es que éste permita repetir las tareas que hayamos programado y para hacer esto se pueden utilizar dos formas tales como:

- `while(1){ }`,
- `for(;;){ }`.

Cualquiera de estas dos instrucciones nos permitirá repetir las tareas de manera infinita ¡claro! no olvidará que éstas se repiten siempre y cuando la tarjeta esté alimentada con corriente eléctrica.

En seguida se muestra el código hecho en Atmel Studio para encender y apagar LEDs durante un cierto tiempo, acá todos los pines del puerto D se configuraron como salidas.

```

1  #include <asf.h>
2  int main (void)
3  {
4      board_init();
5
6      /* Configuramos los puertos a utilizar. */
7      REG_PMC_PCERO = 0x00004000;
8      REG_PIOD_PER  = 0x000007FF;
9      REG_PIOD_PUDR = 0x000007FF;
10     REG_PIOD_OER  = 0x000007FF;
11
12     /* Variables del programa. */
13     int tiempo = 1000;
14
15     /* Esto me permite hacer un loop de manera infinita. */
16     for(;;)
17     {
18         REG_PIOD_SODR = 0x000007FF;
19
20         // Simulamos un tipo retardador
21         for (int i=0; i<tiempo; i++)
22             for (int j=0;j<tiempo; j++){}
23
24         REG_PIOD_CODR = 0x000007FF;
25
26         // Simulamos un tipo retardador
27         for (int i=0; i<tiempo; i++)
28             for (int j=0;j<tiempo; j++){}
29     }
30 }

```

El siguiente código está diseñado para que el usuario que, a través de un interruptor pueda encender LEDs y luego se apaguen, dicho de otro modo, cuando el usuario presione el interruptor los LEDs configurados deberán encenderse y cuando el usuario deje de presionarlo los LEDs deberán de apagarse. Aquí se utilizaron los puertos D y C. Como podemos observar en el código, en el puerto C se configuró el botón interruptor, si observamos la tabla del puerto C de la Fig. 2.2 el pin PC0 no está disponible por lo que se decidió utilizar el pin PC1, y como ya hemos explicado para poder elegir este pin la representación en binario es como $(0010)_2 = (0x02)_{16}$ el cual se configuró como entrada y con la instrucción Pin Data Status Register (PDSR) podemos saber si este pin contiene información. Otro dato importante es que aparecen dos estructuras de decisión las cuales están anidadas y que contienen la misma instrucción, esto pasa debido a que los interruptores suelen dar rebotes, es decir, la corriente no siempre es constante por lo que hacer esto resulta una medida para configurar de manera adecuada los interruptores. Muchos de estos métodos se pueden colocar en una función con el fin de hacer códigos más limpios.

```

main.c
1  #include <asf.h>
2  int main (void)
3  {
4      boart_init();
5
6      /* Configuramos los pines a utilizar del Puerto D. */
7      REG_PMC_PCERO = 0x00004000; // Señal de reloj para el puerto C
8      REG_PIOD_PER = 0x01; // Activamos los puertos correspondientes
9
10     REG_PIOD_OER = 0x01; // Configuramos como salida tales puertos
11     REG_PIOD_PUDR = 0x01; // Desactivamos las resistencias de Pull-up
        de las salidas
12
13
14     /* Configuramos los pines a utilizar del Puerto C. */
15     REG_PMC_PCERO = 0x00002000; // Señal de reloj para el puerto C
16     REG_PIOC_PER = 0x02; // Activamos los puertos correspondientes
17
18     REG_PIOC_ODR = 0x02; // Configuramos como entrada tales puertos
19
20     while(1){
21         // Apagamos el led
22         REG_PIOD_CODR = 0x01;
23
24         if((REG_PIOC_PDSR & 0x0F) != 0x02){
25             retrasar(2);
26             if((REG_PIOC_PDSR & 0x0F) != 0x02){
27                 // Encendemos el led permanentemente
28                 REG_PIOD_SODR = 0x01;
29
30             }
31         }
32     }
33 }

```

2.3. Configuración del módulo DHT22

El circuito de este módulo es simple, en la Fig. 2.3 se muestra de manera visual las conexiones de este módulo. Como se ha mencionado previamente, este módulo devuelve datos de temperatura como de humedad.

El módulo DHT22 tienen cuatro pines, VCC, GND, pin de datos y un pin no conectado que no tiene uso, como se mostró en la Fig. 1.2. Se requiere una resistencia pull-up de 5 a 10 K Ω para mantener la línea de datos alta y para permitir la comunicación entre el sensor y la placa Arduino Due.

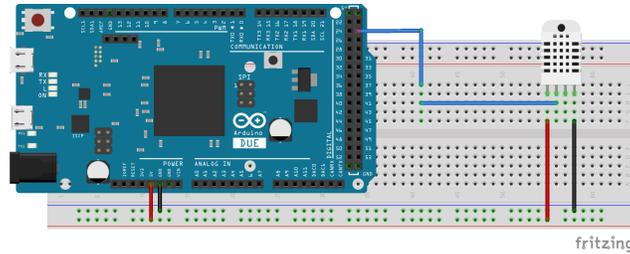


Figura 2.3: Circuito del modulo DHT22 conectado al Arduino Due.

Asimismo, el módulo DHT22 tienen su propio protocolo de un sólo cable utilizado para transferir los datos. Este protocolo requiere una sincronización precisa y los diagramas de sincronización para obtener los datos de los sensores se pueden encontrar en las hojas de datos de los sensores. Sin embargo, no tenemos que preocuparnos mucho por estos diagramas de tiempo, debido a que Arduino diseño una biblioteca DHT que se puede encontrar en el sitio web oficial de Arduino. Dicha biblioteca se incluye en el código para luego definir el número de pin al que está conectado nuestro sensor y crear un objeto DHT. En la sección de configuración se necesita grabar un archivo de textos a un cierto tiempo y ahí grabar los datos del sensor. Usando la función `read22()` se leerán los datos del sensor y colocaremos los valores de la temperatura y la humedad en las variables T y H, junto a ellos el valor de tiempo en la variable t. Al final grabarán los valores de tiempo, temperatura y humedad.

2.4. Configuración del módulo HC-06

El circuito es tan simple y pequeño que sólo se deben realizar algunas conexiones. Se debe de conectar un LED negativo a GND del Arduino y positivo al pin 25 (Puerto D) con una resistencia valorada entre $220\Omega - 1k\Omega$. Con el circuito que se muestra en la Fig. 2.4. **Nota:** No se debe de conectar RX a RX y TX a TX de Bluetooth a Arduino Due, ya que no recibirá datos, aquí TX significa Transmitir y RX significa Recibir.

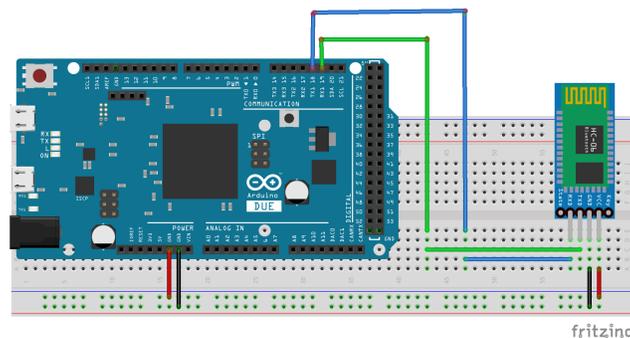


Figura 2.4: Circuito del modulo HC-06 conectado al Arduino Due.

Una vez configurado se procede a hacer la comunicación del Bluetooth HC-06 con el teléfono celular, en el teléfono celular se necesita un aplicación conocida como Serial Bluetooth Terminal. Esta aplicación es muy sencilla ya que sólo necesita conectar por conexión Bluetooth y posterior a ello se le envían comandos a la terminal del Arduino Due, donde los comandos serán interpretados por el microcontrolador y consultará en el código para dar instrucciones o no respecto a lo que se quiera hacer, por ejemplo activar leds, sensores, envío de datos, entre otras funciones. En la Fig. 2.5 se muestran los datos enviados por Arduino Due a través del módulo HC-06. Los primeros datos es para ver las conexión de datos, posteriormente se muestra la conexión Bluetooth que tiene como nombre "Ced01" y respondiendo como la conexión establecida. Una vez hecho esto, la forma en la que configuró fue que al enviarle el valor de 1, este nos devuelve los datos leídos del módulo DHT22, devolviendo la medición de Humedad y Temperatura. El último dato devuelto, acercamos el DHT22 al vapor de una taza de café caliente como prueba, con el fin de ver el funcionamiento del sensor obteniendo un valor esperado para la humedad.



```
10:46:53.280 3
10:46:54.996 3
10:46:55.213 3
10:46:55.395 3
10:46:55.563 3
10:46:55.746 3
10:46:57.368 3567
10:46:57.569 3567
10:46:57.738 3567
10:46:57.937 3567
10:47:01.248 2
10:48:04.395 Connection lost
10:48:10.908 Connecting to Ced01 ...
10:48:13.689 Connected
10:50:43.234 1
10:51:12.892 1
10:51:13.423 Humidity: 48.30 %, Temp: 18.00 Celsius
10:51:23.521 2
10:51:30.226 1
10:51:30.380 Humidity: 48.20 %, Temp: 18.00 Celsius
10:51:45.505 1
10:51:45.841 Humidity: 48.10 %, Temp: 18.00 Celsius
10:51:55.449 1
10:51:55.523 Humidity: 48.10 %, Temp: 18.00 Celsius
10:52:04.978 1
10:52:05.148 Humidity: 48.00 %, Temp: 18.00 Celsius
10:52:10.129 1
10:52:10.188 Humidity: 99.90 %, Temp: 19.90 Celsius
```

Figura 2.5: Resultados enviados del Arduino Due a través del Bluetooth, donde se ven los datos del módulo DHT22.

2.5. Lámpara de LEDs Ultravioleta

La construcción de una lámpara donde la fuente de iluminación son LEDs ofrece ciertas ventajas respecto al utilizar otras bombillas. Un LED cuenta con un bajo consumo de energía y de larga vida duradera.

La lámpara de LEDs ultravioleta se configuró de acuerdo al circuito que se muestra en al Fig. 2.6, esto es una combinación de los circuitos en serie y paralelo. Esta configuración permite ahorrar espacio. Y en la Fig. 2.7 se muestra dicha configuración.

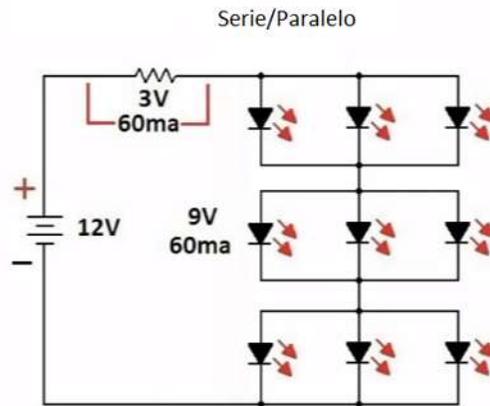


Figura 2.6: Circuito para la construcción de una lámpara LEDs.

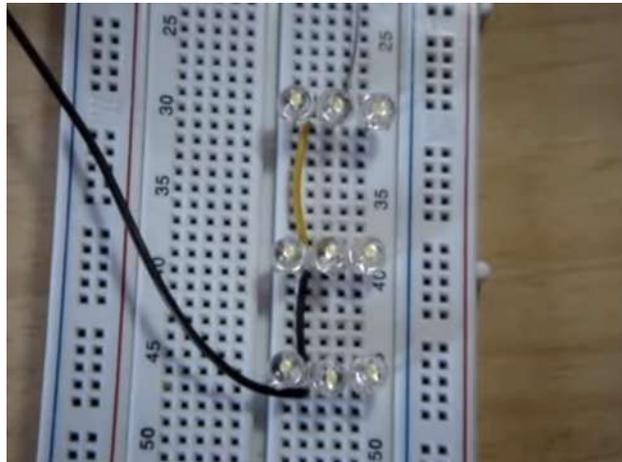


Figura 2.7: Conexión de los LEDs de acuerdo al circuito mostrado en al Fig. 2.6.

Capítulo 3

Métodos numéricos

Las simulaciones por computadora se han convertido en una herramienta de suma importancia hoy en la actualidad ya que podemos realizar simulaciones de problemas físicos, matemáticos, químicos, biológicos, entre otras ramas, de tal forma que nos permite entender si un modelo pueda funcionar. Por ejemplo, el crecimiento de una bacteria, la evolución de tumor en el cerebro o la resistencia de materiales, por mencionar algunos de ellos. Para poder resolver este tipo de problemas es importante conocer ciertas herramientas tales como las que se van a describir en las próximas secciones.

3.1. Métodos introductorios

Los métodos numéricos constituyen procedimientos alternativos de gran utilidad con la finalidad de resolver problemas matemáticos para los cuales se dificulta la utilización de métodos analíticos tradicionales y, ocasionalmente, son la única posible solución. Las aproximaciones son técnicas mediante las cuales un modelo matemático es resuelto usando solamente operaciones aritméticas, tediosos cálculos aritméticos. Son técnicas sistemáticas cuyos resultados son aproximaciones del verdadero valor que asume la variable de interés; la repetición consistente de la técnica, a lo cual se denomina iteraciones, es lo que permite acercarse cada vez más al valor buscado. Por otro lado, se entiende por aproximación numérica x una cifra que representa a un número cuyo valor exacto es X . En la medida en que la cifra x se acerca más al valor exacto x , será una mejor aproximación de ese número.

3.1.1. Método de Euler

El método de Euler es una de las técnicas más elementales de aproximación para resolver problemas de valor inicial, pues la simplicidad de su derivación puede ser utilizada para ilustrar el funcionamiento de otras técnicas más avanzadas. El objetivo del método de Euler sirve para obtener la aproximación a la solución de un problema de valor inicial bien definido

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = y_0. \quad (3.1)$$

Una aproximación continua a la solución $y(t)$ no es posible obtenerla, en lugar de eso, se elige un cierta cantidad de valores la cual se conoce como una **mallá de puntos** que depende del intervalo $[a, b]$ donde esté definido dicho problema. Para esto, primero debemos elegir la cantidad de datos a trabajar, sea N un número entero positivo que definirá la cantidad de datos divididos para cierta variable independiente

$$t_i = a + ih, \quad \text{para cada } i = 0, 1, 2, \dots, N.$$

La distancia entre cada punto está dado por

$$h = \frac{(b - a)}{N} = t_{i+1} - t_i,$$

esto es conocido como el tamaño de paso. Además a todo este conjunto de la construcción de la mallá de puntos se le nombra como la discretización correspondiente al dominio definido.

Utilizando el teorema de Taylor para derivar el método de Euler. Primeramente, supongamos que $y(t)$ representa la única solución de (3.1), el cual contiene dos derivadas continuas sobre el intervalo $[a, b]$ entonces para cada $i = 0, 1, 2, \dots, N - 1$,

$$y_{i+1} = y_i + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2}y''(\xi_i),$$

para algún número ξ_i en (t_i, t_{i+1}) . Como $h = t_{i+1} - t_i$ tenemos

$$y_{i+1} = y_i + hy'(t_i) + \frac{dt^2}{2}y''(\xi_i),$$

y como $y(t)$ satisface la ecuación diferencial (3.1),

$$y_{i+1} = y_i + hf(t_i, y(t_i)) + \frac{dt^2}{2}y''(\xi_i). \quad (3.2)$$

Aquí es donde se construye el método de Euler haciendo un truncamiento tal que $w_i \approx y(t_i)$, para cada $i = 0, 1, \dots, N$ y eliminando el resto de los términos. El método de Euler es

$$w_0 = y_0, \quad (3.3)$$

$$w_{i+1} = w_i + hf(t_i, w_i). \quad (3.4)$$

Para este tipo de problemas bien definidos de valores iniciales donde se conoce la solución exacta del problema, para saber cuánto fue el error producido por el método numérico, basta con calcular la diferencia entre la solución exacta y numérica,

$$e_i = y(t_i) - w_i. \quad (3.5)$$

La Ec. (3.4) es conocida como la **ecuación diferencial** asociada al método de Euler, es la manera de como se construyen la mayoría de las técnicas más avanzadas para aproximar ecuaciones diferenciales de orden n -ésimo. En la Fig. 3.1 se describe la forma en la que se aproxima la solución exacta de manera numérica.

Una vez explicado la teórica de como se derivó el método de Euler, es importante ilustrar con algunos ejemplos. En la Fig. 3.2 se muestran los resultados para el siguiente problema

$$\frac{dy}{dt} = y - t^2 + 1, \quad 0 \leq t \leq 2, \quad y(0) = 0.5.$$

En la Fig. 3.3 se muestran los resultados para el siguiente problema

$$\frac{dy}{dt} = \cos(2t) + \sin(3t), \quad 0 \leq t \leq 1, \quad y(0) = 1.$$

Y por último en la Fig. 3.4 se muestran los resultados para el siguiente problema

$$\frac{dy}{dt} = y/t - (y/t)^2, \quad 1 \leq t \leq 2, \quad y(1) = 1.$$

Cabe mencionar que se realizó para diferentes resoluciones, es decir, se eligieron los valores de $N = 10, 20$ y 40 , con el fin de mostrar que entre más puntos esté dividida la malla se obtendrán mejores resultados, dicho de otro modo, la solución numérica se aproximará mucho más a la solución exacta del problema. En cada Fig. para cada resolución del lado derecho se muestra la gráfica del error, como se observa en cada resolución se obtiene un error cada vez más pequeño.

Para la Fig. 3.2 se comparó con la solución exacta

$$y(t) = (t + 1)^2 - 0.5e^t,$$

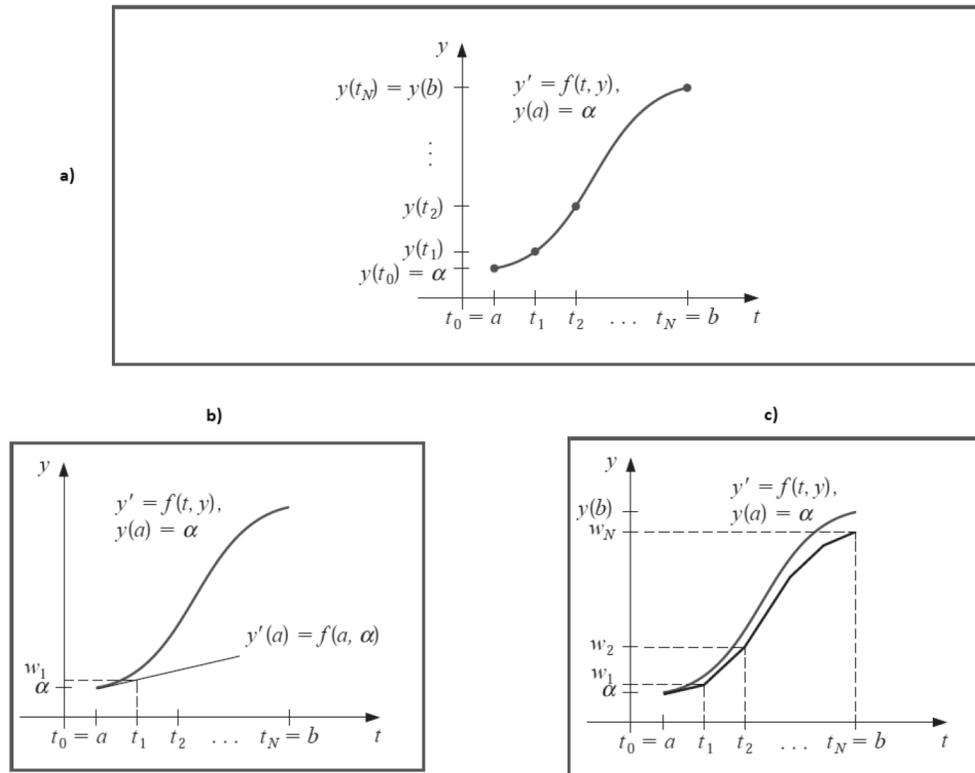


Figura 3.1: a) Se muestra la función aproximar. b) Se muestra un paso para el método de Euler, se observa la pendiente del método. c) Se muestra un conjunto de pendientes para diferentes valores de t_i [29].

mientras que en la Fig. 3.3 la solución exacta

$$y(t) = (1/2)\sin(2t) - (1/3)\cos(3t) + (4/3),$$

y finalmente en la Fig. 3.4 la solución exacta

$$y(t) = \frac{t}{(1 + \ln(t))}.$$

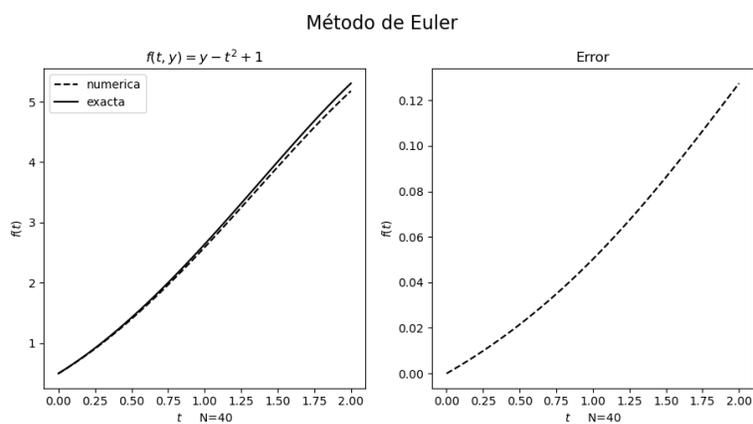
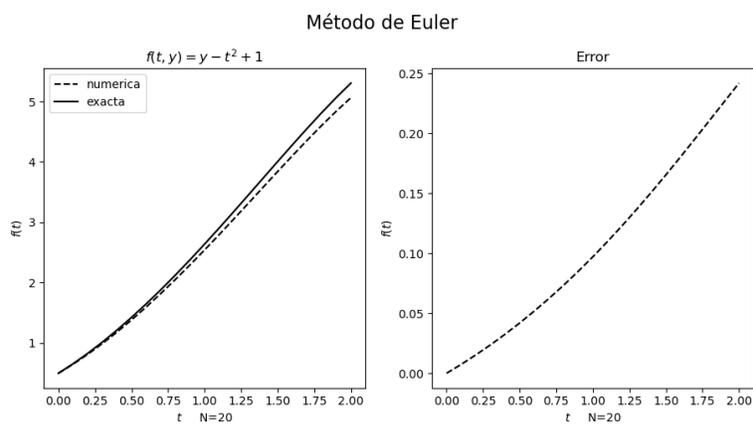
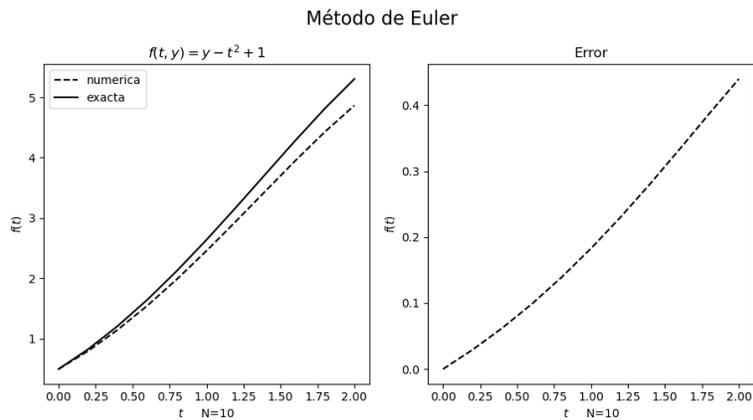


Figura 3.2: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = (t + 1)^2 - 0.5e^t$.

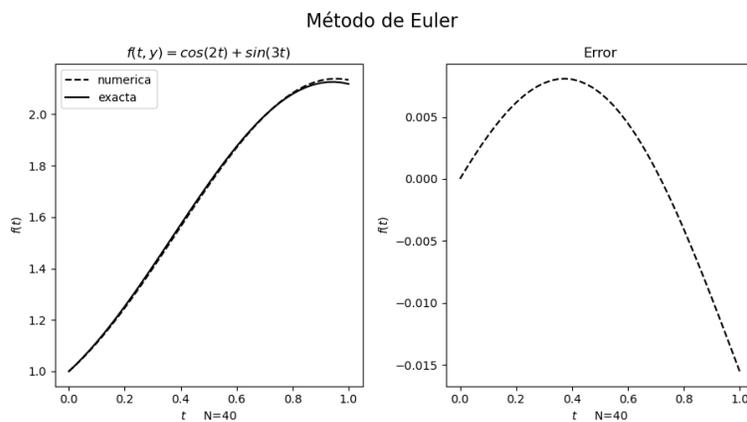
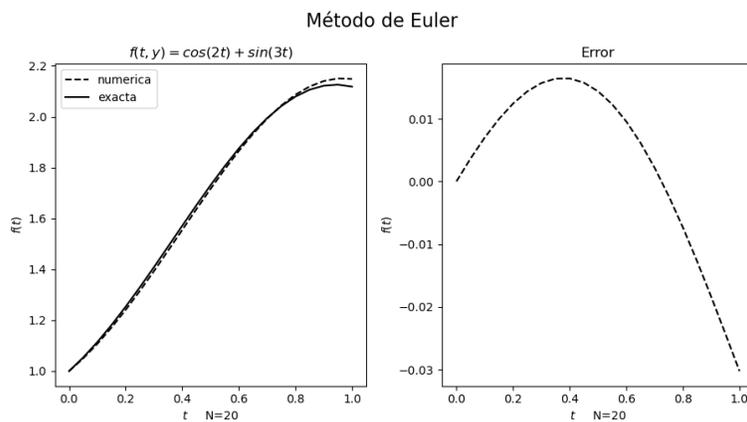
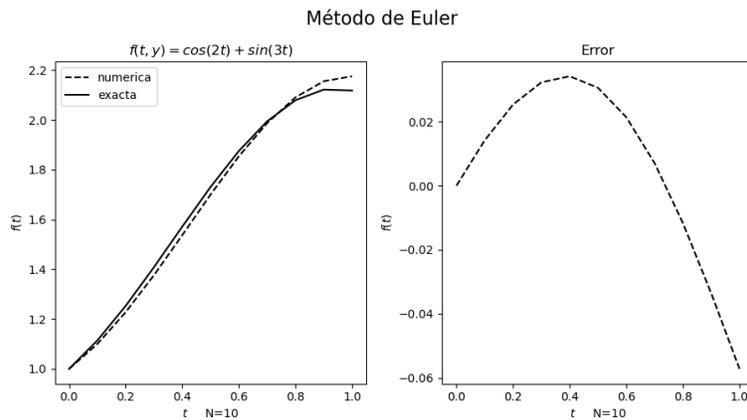


Figura 3.3: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = (1/2)\sin(2t) - (1/3)\cos(3t) + (4/3)$.

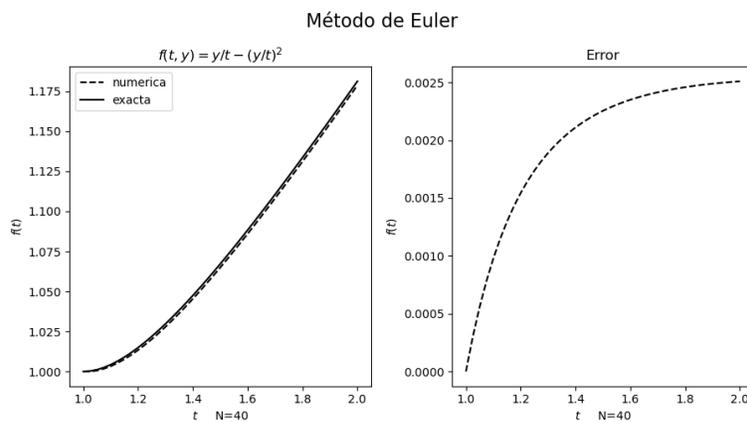
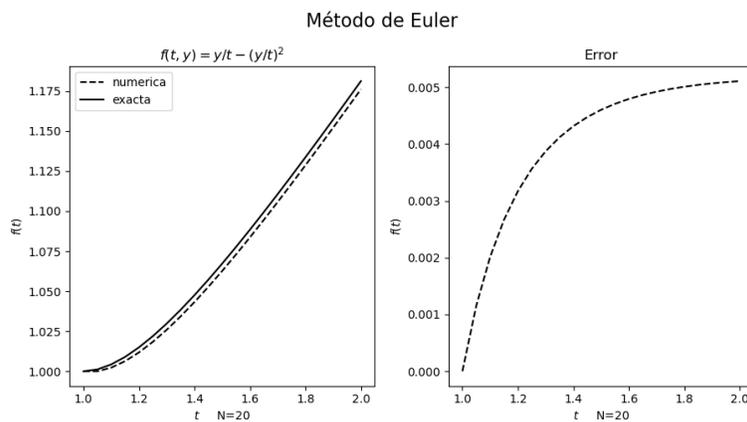
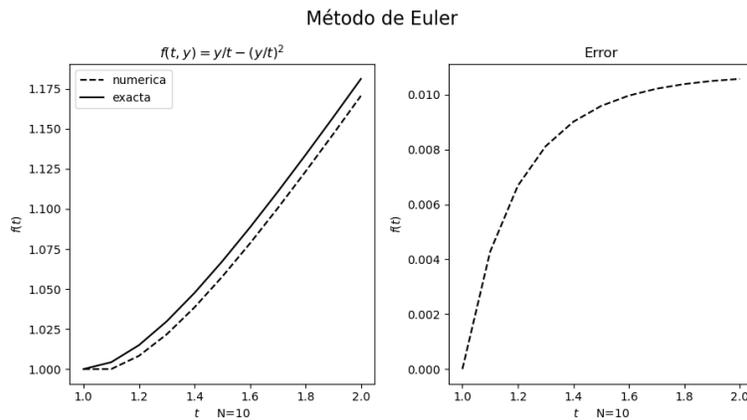


Figura 3.4: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = t/(1 + \ln(t))$.

La forma en la que se derivó el método de Euler es utilizando el teorema de Taylor para derivadas de primer orden, si embargo utilizando esta misma lógica podemos construir un método de aproximación de orden mayor. Este tipo de construcción permite obtener una mejor aproximación a la solución exacta, sin embargo, se tiene que realizar más operaciones como derivar a $f^n(x, y(t))$ n veces por lo que implica más trabajo computacional, además que el error es mucho mayor $O(h^n)$ [30]. En términos generales el método integrador es de la forma

$$\begin{aligned} w_0 &= y_0, \\ w_{i+1} &= w_i + hT^n(t_i, w_i), \quad \text{para cada } i = 0, 1, \dots, N-1, \end{aligned} \quad (3.6)$$

donde

$$T^n(t_i, w_i) = f(t_i, w_i) + \frac{h}{2}f'(t_i, w_i) + \dots + \frac{h^{n-1}}{n!}f^{n-1}(t_i, w_i). \quad (3.7)$$

El método de Euler es el método de Taylor de orden uno.

3.1.2. Métodos Runge Kutta

En la sección anterior se hablo del método de Taylor el cual indica que el error local de truncamiento es de orden $O(h^n)$, además de que es necesario calcular y evaluar las derivadas dependiendo del orden que se haya elegido de Taylor. Esto es complicado y conjuntamente muestra un tiempo de consumo en cuestión de procedimiento para muchos problemas. Por esta razón el método de Taylor es raramente usado en la práctica. Los **métodos Runge Kutta** tiene un error local de truncamiento de orden alto al igual que los métodos de orden n de Taylor pero con la diferencia de que ya no es necesario calcular y evaluar las derivadas de $f(t, y)$.

Fundamentalmente, todos los métodos de Runge Kutta son una generalización básica del método de Euler (Ec. (3.4)) en donde la pendiente es remplazada por un **peso promedio** de las pendientes sobre el intervalo $t_n \leq t \leq t_{n+1}$. Esto es

$$y_{n+1} = y_n + dt \underbrace{(\alpha_1 k_1 + \beta_2 k_2 + \dots + q_m k_m)}_{\text{peso promedio}}. \quad (3.8)$$

Aquí $q_i, i = 1, 2, \dots, m$ son constantes que generalmente satisfacen que $\alpha_1 + \beta_2 + \dots + q_m = 1$ y cada $k_i, i = 1, 2, \dots, m$ es la función f evaluada en un punto elegido (t, y) para cada $t_n \leq t \leq t_{n+1}$. El número m es llamado orden del método. Observando que si se elige a $m = 1, \alpha_1 = 1$ y $k_1 = f(t_n, y_n)$ obtenemos el método de Euler con la forma $y_{n+1} = y_n + hf(t_n, y_n)$. Entonces aquí el método de Euler es conocido como el **método de Runge Kutta de primer orden**.

Teorema 1 Supongamos que $f(t, y)$ y todas su derivadas parciales de orden menor que $n + 1$ son continuas en $D = \{(t, y) | a \leq t \leq b, c \leq y \leq d\}$, y dado a $(t_0, y_0) \in D$. Para todo $(t, y) \in D$, existe un ξ entre t y t_0 y μ entre y e y_0 con

$$f(t, y) = P_n(t, y) + R_n(t, y),$$

donde

$$P_n(t, y) = f(t_0, y_0) + \left[\frac{1}{n!} \sum_{j=0}^n \binom{n}{j} (t - t_0)^{n-j} (y - y_0)^j \frac{\partial^n}{\partial t^{n-j} \partial y^j} f(t_0, y_0) \right]$$

y

$$R_n(t, y) = \frac{1}{(n + 1)!} \sum_{j=0}^{n+1} \binom{n + 1}{j} (t - t_0)^{n+1-j} (y - y_0)^j \frac{\partial^n}{\partial t^{n+1-j} \partial y^j} f(\xi, \mu).$$

La función $P_n(t, y)$ es conocida como el **polinomio de Taylor n -ésimo en dos variables** para la función f en el punto (t_0, y_0) , y $R_n(t, y)$ es el resto asociado con $P_n(t, y)$.

La demostración de este teorema 1 se puede encontrar en distintos libros de cálculo [31]. Prácticamente el teorema 1, nos permite deshacernos del cálculo y evaluación de las derivadas que es lo que pasaba con los métodos de Taylor. El primer paso para derivar el método de Runge Kutta es para determinar los valores de a_1, α_1 y β_1 con la propiedad de que $a_1 f(t + \alpha_1, y + \beta_1)$ aproxime a

$$T^{(2)}(t, y) = f(t, y) + \frac{h}{2} f'(t, y),$$

con un error de aproximación no mayor que $O(h^2)$, el cual representa como se presenta el orden y error local de truncamiento del método de Taylor de orden dos. Entonces,

$$f'(t, y) = \frac{d}{dt} f(t, y) = \frac{\partial}{\partial t} f(t, y) + \frac{\partial}{\partial y} f(t, y) \cdot y'(t) \quad \text{con} \quad y'(t) = f(t, y),$$

sustituyendo en el método de Taylor se tiene

$$T^{(2)}(t, y) = f(t, y) + \frac{h}{2} \frac{\partial}{\partial t} f(t, y) + \frac{h}{2} \frac{\partial}{\partial y} f(t, y) \cdot f(t, y). \quad (3.9)$$

Haciendo uso del teorema 1 para expandir a $a_1 f(t + \alpha_1, y + \beta_1)$ como se había comentado, esto para el orden uno en el punto (t, y) lo que da a resolver

$$a_1 f(t + \alpha_1, y + \beta_1) = a_1 P_1(t + \alpha_1, y + \beta_1) + a_1 R_1(t + \alpha_1, y + \beta_1),$$

donde $P_1(t + \alpha_1, y + \beta_1)$ resulta

$$P_1(t + \alpha_1, y + \beta_1) = a_1 f(y, t) + a_1 \alpha_1 \frac{\partial}{\partial t} f(t, y) + a_1 \beta_1 \frac{\partial}{\partial y} f(t, y), \quad (3.10)$$

y $R_1(t + \alpha_1, y + \beta_1)$ nos da

$$R_1(t + \alpha_1, y + \beta_1) = \frac{\alpha_1^2}{2} \frac{\partial^2}{\partial t^2} f(\xi, \mu) + \alpha_1 \beta_1 \frac{\partial^2}{\partial t \partial y} f(\xi, \mu) + \frac{\beta_1^2}{2} \frac{\partial^2}{\partial y^2} f(\xi, \mu), \quad (3.11)$$

para algún ξ entre t y $t + \alpha_1$ y μ entre y y $y + \beta_1$.

Relacionando los coeficientes de f y sus derivadas calculadas dentro de las ecuaciones (3.9) y (3.10) nos da tres ecuaciones

$$f(t, y) : a_1 = 1; \quad \frac{\partial}{\partial t} f(t, y) : a_1 \alpha_1 = \frac{h}{2}; \quad y \quad \frac{\partial}{\partial y} f(t, y) : a_1 \beta_1 = \frac{h}{2} f(t, y).$$

Resolviendo para los tres parámetros a_1 , α_1 y β_1 son por lo tanto

$$a_1 = 1, \quad \alpha_1 = \frac{h}{2}, \quad \beta_1 = \frac{h}{2} f(t, y),$$

entonces podemos escribir a

$$T^{(2)}(t, y) = f\left(t + \frac{h}{2}, y + \frac{h}{2} f(t, y)\right) - R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t, y)\right),$$

y de la Ec. (3.11)

$$R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t, y)\right) = \frac{h^2}{8} \frac{\partial^2}{\partial t^2} f(\xi, \mu) + \frac{h^2}{4} f(t, y) \frac{\partial^2}{\partial t \partial y} + \frac{h^2}{8} (f(t, y))^2 \frac{\partial^2}{\partial y^2} f(\xi, \mu).$$

Si todas las derivadas parciales de f están en el límite, entonces

$$R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t, y)\right)$$

es de $O(h^2)$. Como consecuencia se tiene *el orden del error para este nuevo método es el mismo que el método de Taylor de orden dos*. El método de la ecuación diferencial resulta de reemplazar a $T^2(t, y)$ en el método de Taylor de orden dos por $f(t + (h/2), y + (h/2)f(t, y))$ que a su vez es el método de **Runge Kutta 2** es conocido de como **método del punto medio**.

Finalmente el integrador resulta de la siguiente forma

$$w_0 = y_0 \quad (3.12)$$

$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right), \quad (3.13)$$

todo esto para $i = 0, 1, 2, \dots, N - 1$. Como podemos observar sólo tres de los parámetros son presentados en $a_1 f(t + \alpha_1, y + \beta_1)$ y todos necesitan estar relacionados con $T^{(2)}$.

En las figuras 3.5, 3.6 y 3.7 se muestran los ejemplos realizados para el integrador numérico *método del punto medio* o *runge kutta 2*. Estos ejemplos son los mismos que se resolvieron haciendo uso del método de Euler. Es bastante claro que el método de Runge Kutta 2 posee un error mucho más pequeño que el método de Euler, por ejemplo si comparamos la Fig. 3.5 con la Fig. 3.2 la aproximación a la solución exacta por parte del Runge Kutta 2 es mucho más exacta que el método de Euler, lo mismo pasa con los otros dos ejemplos.

Otra forma de escribir el *método del punto medio* haciendo referencia a la Ec. (3.8) quedando de la siguiente forma

$$w_0 = 0,$$

$$w_{i+1} = w_i + \frac{h}{2}(k_1 + k_2),$$

donde

$$k_1 = f(t_i, w_i) \quad \text{y} \quad k_2 = f(t_i + h, w_i + hk_1).$$

Otro método a estudiar es el **Método de Runge Kutta de orden cuatro** el cual se construye de la misma forma que el método de Runge Kutta de orden dos analizado anteriormente. Este método comúnmente es el más utilizado en la forma de la Ec. (3.8) tal como

$$w_0 = y_0,$$

$$k_1 = hf(t_i, w_i),$$

$$k_2 = hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right),$$

$$k_3 = hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right),$$

$$k_4 = hf(t_i, w_i + k_3),$$

$$w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

para cada $i = 0, 1, \dots, N - 1$. Esto es el método del Runge Kutta de orden cuatro y el cual tiene un error local de truncamiento de orden $O(h^4)$, el cual aproxima a la solución $y(t)$ del problema y cuenta con cinco derivadas continuas.

En las figuras 3.8, 3.9 y 3.10 se resuelven los mismos ejercicios que se han resuelto para los métodos integradores de Euler y Runge Kutta de orden dos.

Los métodos de Runge Kutta destacan por la evaluación de f . En los métodos de segundo orden, el error local de truncamiento es $O(h^2)$ y el costo está en evaluar dos funciones por cada paso k_1 y k_2 . Para el método de Runge Kutta de orden cuatro, éste requiere de evaluar a 4 funciones por cada paso y el error local de truncamiento es de $O(h^4)$. En la

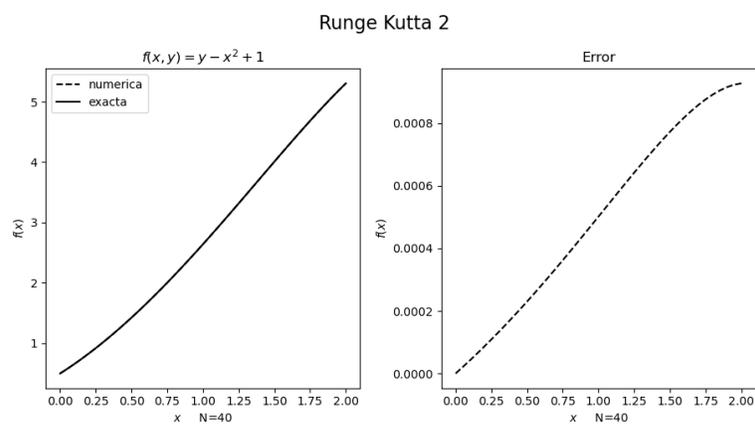
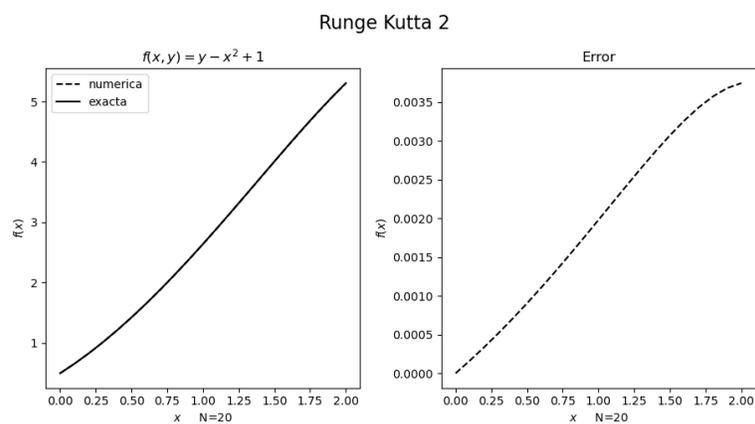
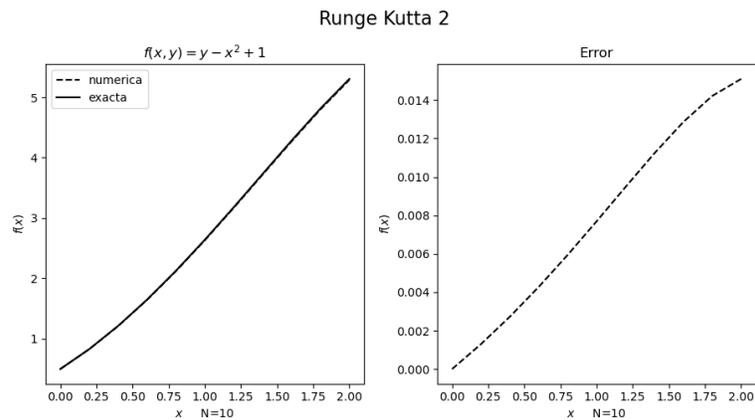


Figura 3.5: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = (t + 1)^2 - 0.5e^t$.

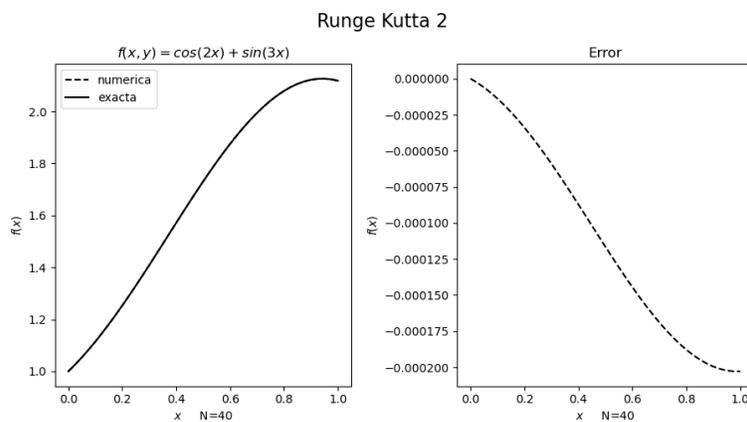
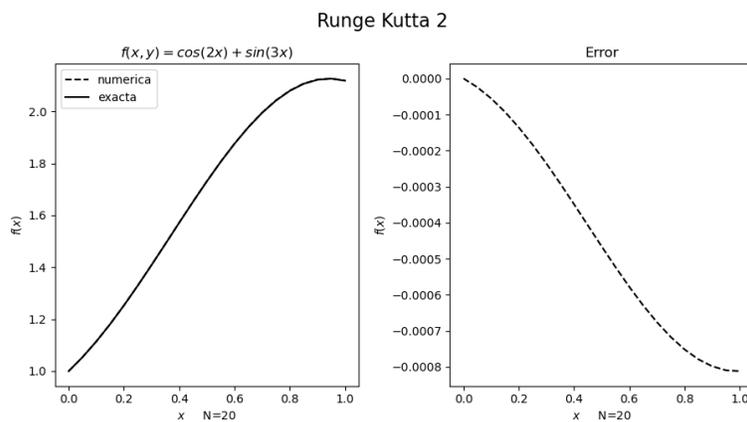
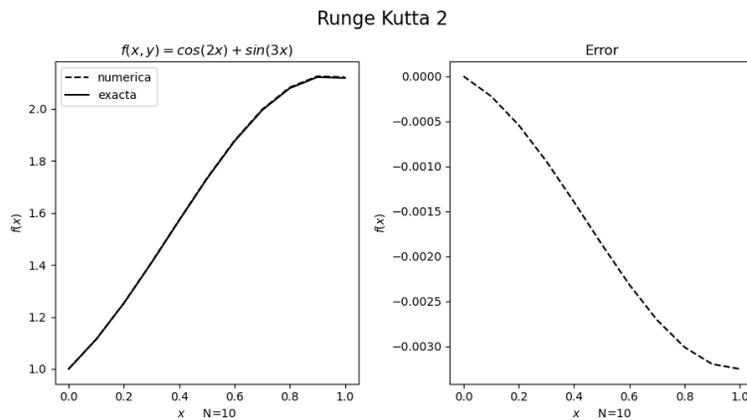


Figura 3.6: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = (1/2)\sin(2t) - (1/3)\cos(3t) + (4/3)$.

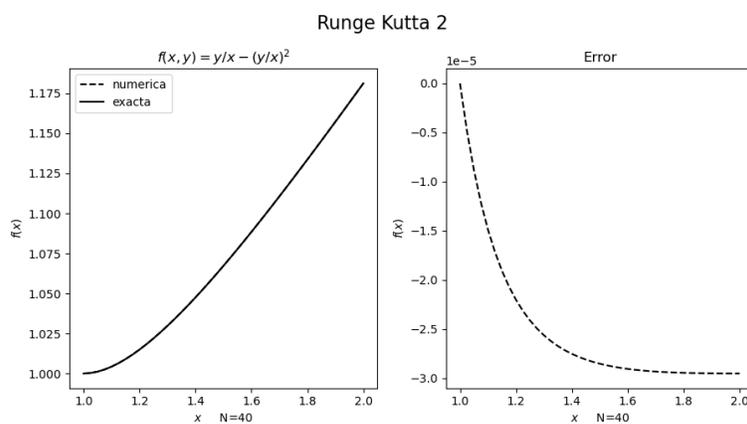
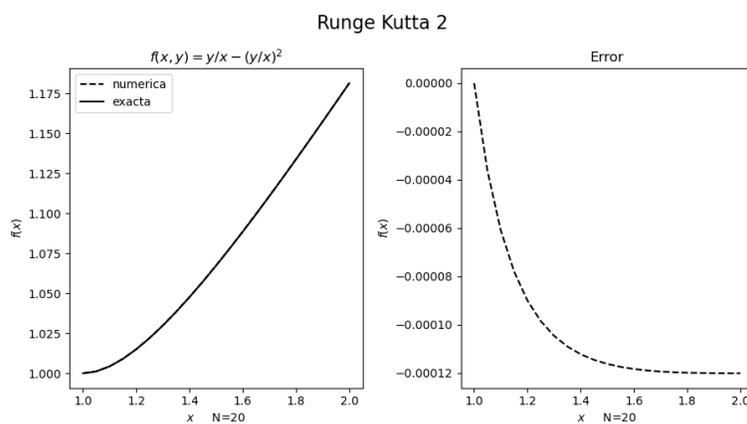
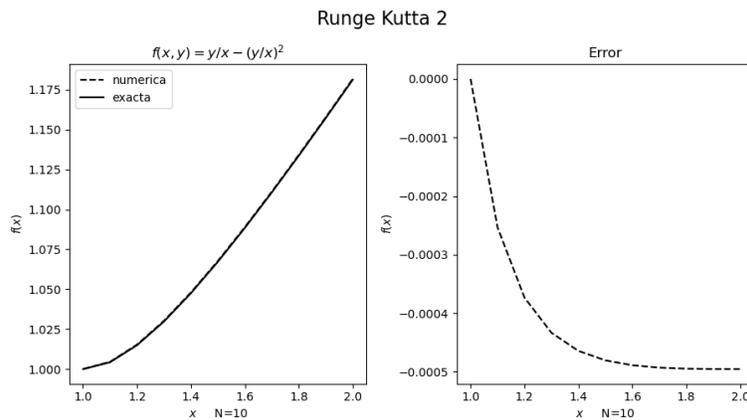


Figura 3.7: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = t/(1 + \ln(t))$.

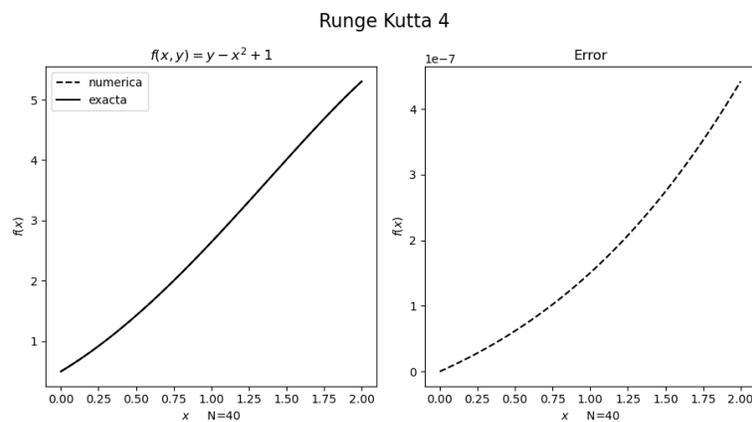
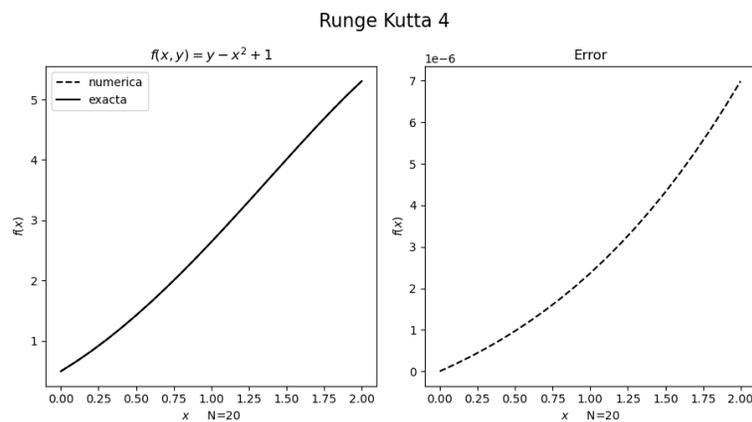
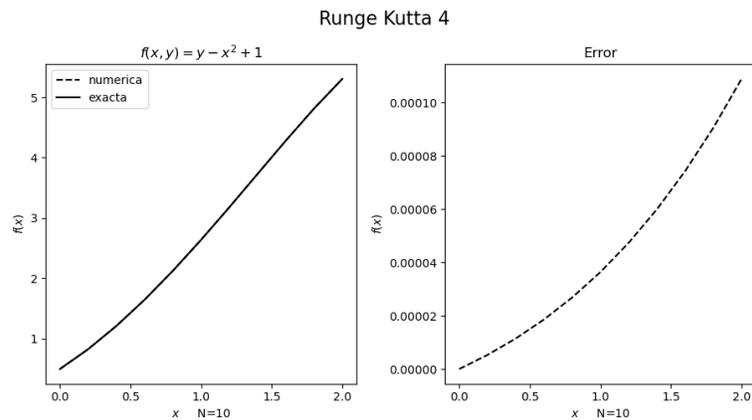


Figura 3.8: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = (t + 1)^2 - 0.5e^t$.

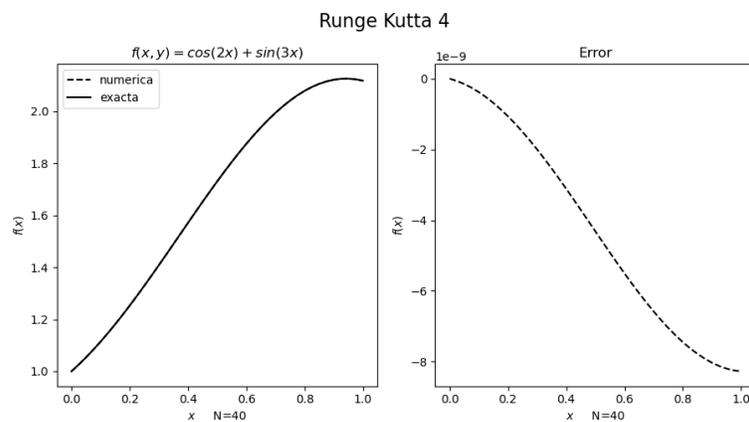
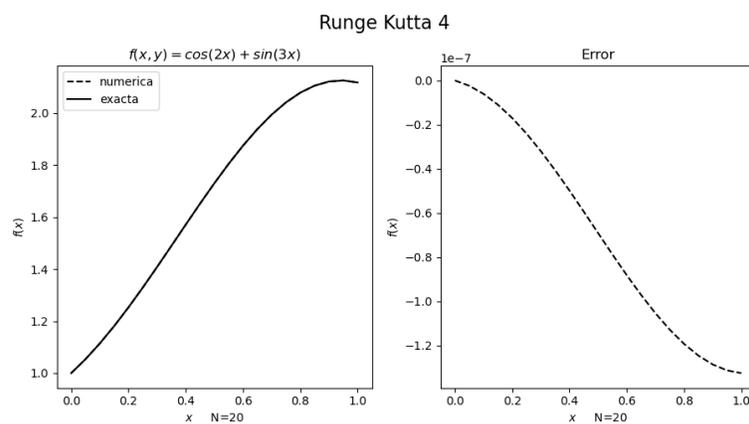
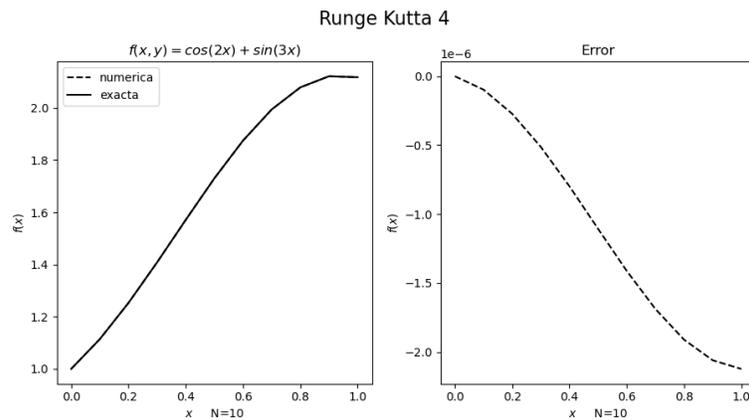


Figura 3.9: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = (1/2)\sin(2t) - (1/3)\cos(3t) + (4/3)$.

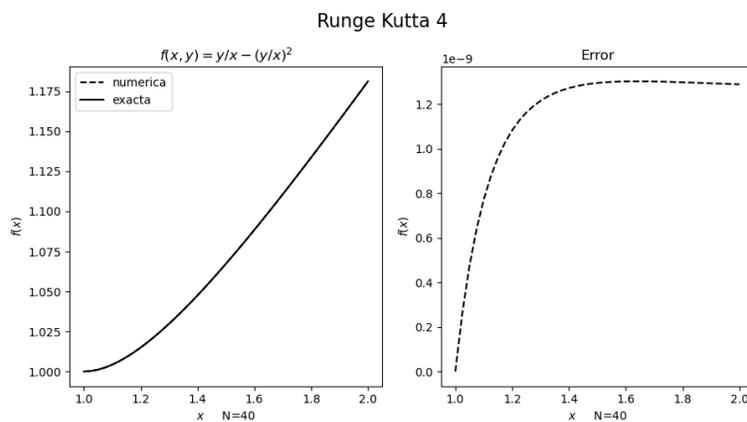
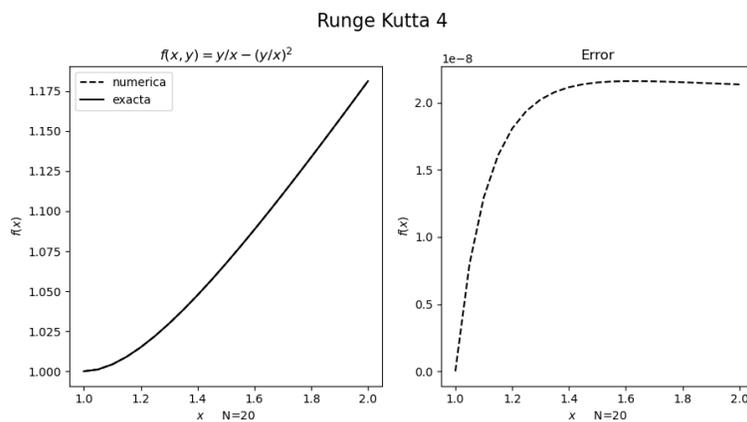
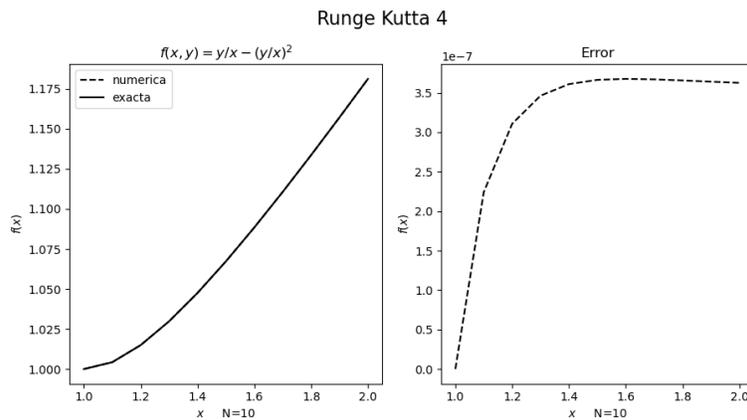


Figura 3.10: Cada gráfica de cada inciso muestra la solución exacta y la solución numérica del lado izquierdo mientras que del lado derecho muestra el error al compararse con la solución exacta $y(t) = t/(1 + \ln(t))$.

tabla 3.1 el autor Butcher ha establecido la relación entre el número de evaluaciones por cada paso y el orden del error local de truncamiento. En esta tabla se explica porque los métodos de orden menor a cinco con un tamaño de paso h más pequeño es requerido a diferencia de lo métodos de orden mayor que sólo necesitan un tamaño de paso h un poco mayor.

Evaluación por paso	2	3	4	$5 \leq n \leq 7$	$8 \leq n \leq 8$	$10 \leq n$
Mejor posible error local de truncamiento	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^{n-3})$

Tabla 3.1: Evaluación por cada paso para un error local de truncamiento [32].

La tabla 3.1 podemos comprenderla de la siguiente manera: el método de Runge Kutta de orden cuatro requiere de cuatro evaluaciones por cada paso, mientras que el método de Euler requiere sólo de una evaluación por cada paso. Por lo tanto, esto significa que el método de Runge Kutta de orden cuatro nos regrese un mejor resultado aproximado a la solución exacta del problema que el método de Euler. De manera similar el método de Runge Kutta de orden cuatro es superior al método de Runge Kutta de orden dos el cual requiere de dos evaluaciones por cada paso. Esto debería de ser más exacto a la solución exacta con tamaño de paso h que un método de orden dos con tamaño de paso $h/2$.

A diferencia de los métodos estudiados anteriormente y aquellos que no mencionamos como el *método de Heun* donde se elije a $T^{(3)}$ con la expresión de la forma $f(t + \alpha_1, y + \delta_1 f(t + \alpha_2, y + \delta_2 f(t, y)))$ el cual involucra a cuatro parámetros, el método de Runge Kutta 4 es el más utilizado. Veamos la siguiente tabla 3.2 donde se compara el resultado de los integradores programados. Para el método de Euler se elige a $h = 0.025$, para el método de Runge Kutta 2 $h = 0.05$ y para el método del Runge Kutta 4 tomamos a $h = 0.1$ los cuales son comparados con la malla de puntos $t_i = 0.1, 0.2, 0.3, 0.4, 0.5$. Al elegir estos puntos hacemos que cada uno de los métodos requiera evaluar a 20 funciones para determinar los valores de la tabla 3.2 para aproximar a $y(0.5)$ para el problema

$$y' = y - t^2 + 1, \quad 0 \leq t \leq 2, \quad y(0) = 0.5.$$

t_i	Exacta	Euler $h = 0.025$	Punto medio $h = 0.05$	Runge Kutta Orden Cuatro $h = 0.1$
0.0	0.5000	0.5000	0.5000	0.5000
0.1	0.6574	0.6554	0.6573	0.6574
0.2	0.8292	0.8253	0.8290	0.8292
0.3	1.0150	1.0089	1.0147	1.0150
0.4	1.2140	1.2056	1.2136	1.2140
0.5	1.4256	1.4147	1.4250	1.4256

Tabla 3.2: Se muestran los valores correspondientes al método de Euler, Punto medio y Runge Kutta 4 [33].

Como es evidente apreciar en la tabla 3.2 el método de Runge Kutta 4 es mejor y superior a los otros métodos.

3.2. Diferencias finitas

Uno de los problemas a resolver en esta tesis es la ecuación de reacción-difusión descrita en la Ec. (1.1) que se describió en capítulo 1, sección 1.2. Como se observa, en esta ecuación involucra derivadas parciales por lo que es necesario programar un método para resolverlas. Cuando se habla de ecuaciones parciales podemos encontrar derivadas del tipo elípticas, parabólicas e hiperbólicas, además comúnmente las ecuaciones diferenciales parciales se categorizan de una manera similar dentro de una sección cónica.

Ecuaciones Elípticas: Son ecuaciones que involucran a derivadas del tipo $u_{xx}(x, y) + u_{yy}(x, y)$ y es una ecuación elíptica. Una de las ecuaciones elípticas conocidas es llamada como la **ecuación de Poisson** que tiene la forma

$$\frac{\partial^2}{\partial x^2}u(x, y) + \frac{\partial^2}{\partial y^2}u(x, y) = f(x, y),$$

donde se asume que f describe la entrada al problema sobre un plano R y frontera S . En este tipo de ecuación se planteado cuando la variable del tiempo t es independiente a problemas físicos por ejemplo la energía potencial de un punto en el plano actúa sobre fuerzas gravitatorias en el plano. Cuando $f(x, y) \equiv 0$, esto se reduce a la **ecuación de Laplace** [34].

Ecuaciones parabólicas: Las ecuaciones parabólicas tienen la forma

$$\frac{\partial}{\partial t}u(x, t) - \alpha^2 \frac{\partial^2}{\partial x^2}u(x, t) = 0,$$

para esto, un problema físico que se considera es el flujo de calor a lo largo de una varilla de longitud l el cual tiene una temperatura uniforme. Por otro lado se requiere que la varilla este perfectamente aislada sobre las superficies laterales. Típicamente, se suele especificar el calor inicial de distribución sobre la varilla de la forma

$$u(x, 0) = f(x),$$

y para describir el comportamiento en el inicio y fin de la varilla. Por ejemplo, si al inicio y final de la varilla se tiene una temperatura constante U_1 y U_2 , entonces las condiciones de frontera se dan como

$$u(0, t) = U_1, \quad \text{y} \quad u(l, t) = U_2,$$

y la distribución de calor tiende la distribución de temperatura como

$$\lim_{t \rightarrow \infty} u(x, t) = U_1 + \frac{U_2 - U_1}{l}x.$$

Si en lugar de que la varilla se encuentre aislada tal que el calor no fluya a lo largo de la varilla, entonces las condiciones de frontera son

$$\frac{\partial}{\partial x}u(0, t) = 0 \quad \text{y} \quad \frac{\partial}{\partial x}u(l, t) = 0.$$

Entonces el calor no escapa de la varilla y en un caso limitado la temperatura es constante sobre la varilla. Las ecuaciones diferenciales parciales son además de suma importancia en el estudio de la difusión de gas; de hecho, esto es conocido como una **ecuación de difusión** [35].

La **ecuación de onda** es un ejemplo de una ecuación diferencial parcial hiperbólica. Supongamos que se tiene una cadena elástica de longitud l la cual es estrechada entre dos soportes en el mismo nivel horizontal. Si la cadena se hace que vibre en un plano vertical, entonces el desplazamiento vertical $u(x, t)$ de un punto x en el tiempo t satisface la ecuación diferencial parcial

$$a^2 \frac{\partial^2}{\partial x^2}u(x, t) - \frac{\partial^2}{\partial t^2}u(x, t) = 0, \quad 0 \leq x \leq l \quad \text{y} \quad 0 \leq t,$$

produciendo efectos de amortiguación ignorados y la amplitud no es demasiado larga. Por lo que al poner restricciones para este problema se asume que la posición y velocidad inicial de la cadena están dadas por

$$u(x, 0) = f(x) \quad \text{y} \quad \frac{\partial}{\partial t}u(x, 0) = g(x), \quad \text{para } 0 \leq x \leq l.$$

Los puntos finales son fijados, y además tenemos que $u(0, t) = 0$ y $u(l, t) = 0$ [36].

Sabiendo las tres formas que se tiene respecto a las ecuaciones diferenciales parciales, la ecuación parcial a la que más se adapta a la ecuación 1.1 son las ecuaciones diferenciales parciales la cual es necesario que la función $u(x, t)$ evolucione respecto al tiempo. Para resolver una ecuación diferencial parcial hiperbólica utilizaremos un método conocido como **diferencias finitas**.

Primeramente debemos de construir nuestro dominio y en la Fig. 3.11 se muestra el par ordenado de puntos (x, t) . Primeramente debemos elegir el tamaño de paso para el par ordenado. Sea n y m los números enteros que definen el tamaño de paso $h = (b - a)/n$ y $k = (d - c)/m$. Cada partición corresponde al intervalo $[a, b]$ que se divide en n partes con tamaño h y el intervalo $[c, d]$ que se divide en m partes con un tamaño m .

De esta forma se genera una cuadrícula sobre el rectángulo R de tal forma que al dibujar tanto líneas verticales como horizontales a través de los puntos con coordenadas (x_i, t_j) donde

$$x_i = a + ih, \quad i = 0, 1, \dots, n \quad \text{y} \quad t_j = c + jk, \quad j = 0, 1, \dots, m.$$

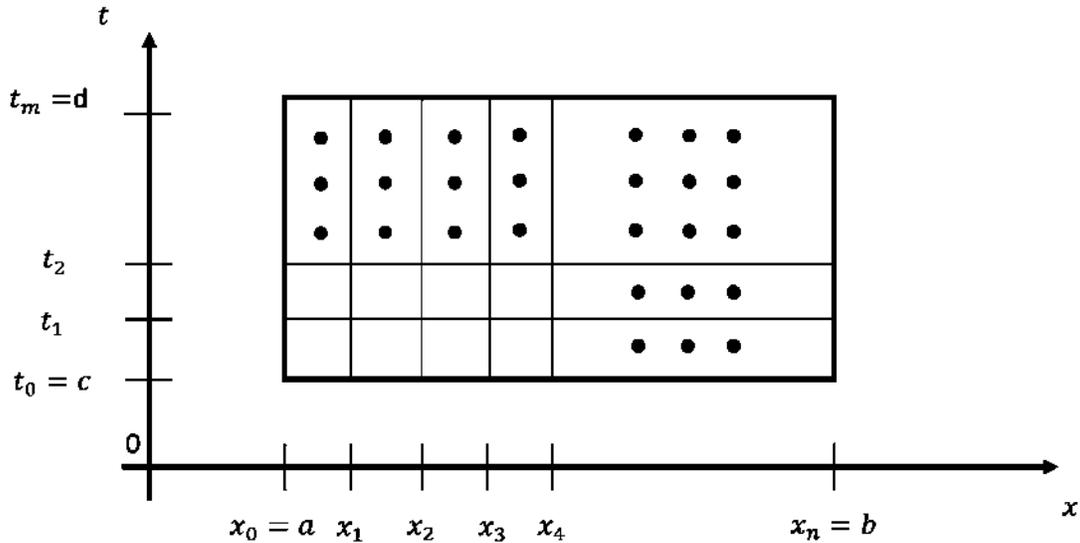


Figura 3.11: Malla de puntos que representa el problema bidimensional.

Las líneas $x = x_i$ y $t = t_j$ forman una cuadrícula y cada una de las intersecciones forma una **malla de puntos** de la cuadrícula. Para cada punto de la malla en el interior de la cuadrícula (x_i, t_j) .

No obstante el problema a resolver con diferencias finitas es la ecuación de onda

$$\alpha^2 \frac{\partial^2}{\partial x^2} u(x, t) - \frac{\partial^2}{\partial t^2} u(x, t) = 0, \quad 0 \leq x \leq l \quad \text{y} \quad 0 < t, \quad (3.14)$$

sujeto a las condiciones

$$\begin{aligned} u(0, t) = u(l, t) = 0, & \quad \text{para} \quad t > 0, \\ u(x, 0) = f(x) \quad \text{y} \quad \frac{\partial}{\partial t} u(x, 0) = g(x), & \quad \text{para} \quad 0 \leq x \leq l. \end{aligned}$$

Tomando a $m > 0$ para definir la cuadrícula del eje x y dividirla en los puntos del tamaño $h = l/m$ y seleccionando el paso $k > 0$ para el tiempo. La malla de puntos (x_i, t_j) son definidos

$$x_i = ih, \quad \text{y} \quad t_j = jk,$$

para cada $i = 0, 1, \dots, m$ y $j = 0, 1, \dots$

Para algún punto en el interior de la malla la ecuación de onda se convierte en

$$\alpha^2 \frac{\partial^2}{\partial x^2} u(x_i, t_j) - \frac{\partial^2}{\partial t^2} u(x_i, t_j) = 0. \quad (3.15)$$

Para obtener el método de diferenciación partimos por expandir a la variable dependiente u para el quinto término del polinomio de Taylor primero para el punto x_i evaluado en el punto x_{i-1} y x_{i+1} , además se asume que $u \in C^4[x_{i-1}, x_{i+1}]$,

$$\begin{aligned} u(x_{i-1}, t_j) &= u(x_i - h, t_j) \\ &= u(x_i, t_j) - hu'(x_i, t_j) + \frac{h^2}{2}u''(x_i, t_j) - \frac{h^3}{6}u'''(x_i, t_j) + \frac{h^4}{24}u^{(4)}(\xi_i^-, t_j), \end{aligned}$$

para algún ξ_i^- dentro de (x_{i-1}, x_i) y

$$\begin{aligned} u(x_{i+1}, t_j) &= u(x_i + h, t_j) \\ &= u(x_i, t_j) + hu'(x_i, t_j) + \frac{h^2}{2}u''(x_i, t_j) + \frac{h^3}{6}u'''(x_i, t_j) + \frac{h^4}{24}u^{(4)}(\xi_i^+, t_j), \end{aligned}$$

para algún ξ_i^+ dentro de (x_i, x_{i+1}) . Ahora sumamos esas dos ecuaciones y resolviendo aritméticamente tenemos

$$u(x_{i-1}, t_j) + u(x_{i+1}, t_j) = 2u(x_i, t_j) + h^2u''(x_i, t_j) + \frac{h^4}{24}[u^{(4)}(\xi_i^-, t_j) + u^{(4)}(\xi_i^+, t_j)]$$

y resolviendo para $u''(x_i, t_j)$ resulta

$$u''(x_i, t_j) = \frac{1}{h^2}[u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)] - \frac{h^4}{24}[u^{(4)}(\xi_i^-, t_j) + u^{(4)}(\xi_i^+, t_j)].$$

Este es un resultado bastante bueno, debido a que se puede simplificar con el teorema del valor medio el cual nos simplificará el término del error, dando

$$u''(x_i, t_j) = \frac{1}{k^2}[u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)] - \frac{k^2}{12}u^{(4)}(\xi_i, t_j), \quad (3.16)$$

donde $\xi_i \in (x_{i-1}, x_{i+1})$. Ahora resolviendo lo mismo pero para la variable independiente t evaluado en el punto t_{j-i} y t_{j+1} asumiendo que $u \in C^4[t_{i-1}, t_{i+1}]$ nos da como resultado

$$u''(x_i, t_j) = \frac{1}{h^2}[u(x_i, t_{j-1}) - 2u(x_i, t_j) + u(x_i, t_{j+1})] - \frac{h^2}{12}u^{(4)}(x_i, \mu_j), \quad (3.17)$$

donde $\mu \in (t_{j-1}, t_{j+1})$. A las ecuaciones (3.16) y (3.17), se les conoce como **cociente de diferencia centrada**. Estas dos ecuaciones se sustituyen en la ecuación de onda (Ec. (3.14)) quedando de la siguiente forma

$$\frac{u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)}{k^2} - \alpha^2 \frac{u(x_i, t_{j-1}) - 2u(x_i, t_j) + u(x_i, t_{j+1}))}{h^2} = 0$$

Omitiendo el término del error

$$\tau_{i,j} = \frac{1}{12} \left[k^2 \frac{\partial^2}{\partial t^4} u(x_i, \mu_j) + \alpha^2 h^2 \frac{\partial^2}{\partial t^4} u(\xi_i, t_j) \right], \quad (3.18)$$

nos queda como resultado a la ecuación diferencial siendo que $w_{i,j} \approx u(x_i, t_j)$

$$\frac{w_{i-1,j} - 2w_{i,j} + w_{i+1,j}}{k^2} - \alpha^2 \frac{w_{i,j-1} - 2w_{i,j} + w_{i,j+1}}{h^2} = 0.$$

Si definimos a $\lambda = \alpha k/h$ entonces podemos escribir de la siguiente forma

$$w_{i-1,j} - 2w_{i,j} + w_{i+1,j} - \lambda^2 w_{i,j-1} - 2\lambda w_{i,j} - \lambda^2 w_{i,j+1} = 0,$$

y resolviendo para $w_{i,j+1}$ se tiene por tanto

$$w_{i,j+1} = 2(1 - \lambda^2)w_{i,j} + \lambda^2(w_{i+1,j} + w_{i-1,j}) - w_{i,j-1}. \quad (3.19)$$

Esta ecuación se puede fijar para cada $i = 1, \dots, m - 1$ y $j = 1, 2, \dots$. Las condiciones de frontera están dadas por

$$w_{0,j} = w_{m,j} = 0, \quad \text{para cada } j = 1, 2, \dots, \quad (3.20)$$

y las condiciones iniciales

$$w_{i,0} = f(x), \quad \text{para cada } i = 1, 2, \dots, m - 1. \quad (3.21)$$

En la Fig. 3.12 se muestra la forma en la que se aproxima el siguiente valor de la ecuación. Los puntos que están en negro representan las condiciones iniciales y las X los valores que toma el método de diferencias centrales para obtener el valor de X en color azul.

Finalmente para resolver un problema del tipo de ecuación de onda se implementa las ecuaciones (3.19), (3.20) y (3.21).

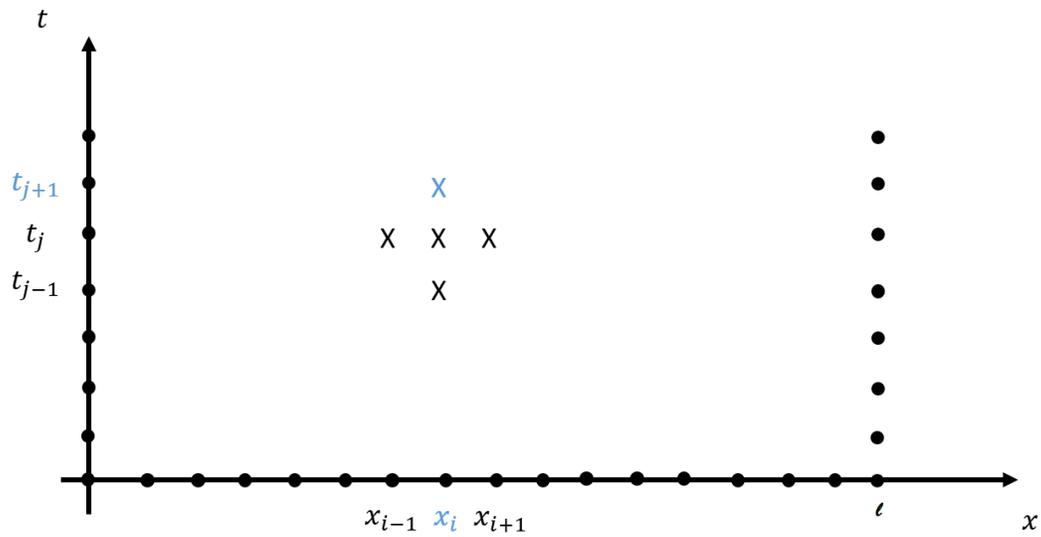


Figura 3.12: Ésta es la forma en la que aproxima los siguientes puntos. Aquí los puntos negros representan las condiciones iniciales.

Problema 1. En la Fig. 3.13 se muestran los resultados al resolver el siguiente problema

$$\frac{\partial^2}{\partial t^2}u(x,t) - 4\frac{\partial^2}{\partial x^2}u(x,t) = 0, \quad 0 < x < 1, \quad 0 < t,$$

con las condiciones de frontera

$$u(0,t) = u(1,t) = 0, \quad \text{para } 0 < t,$$

y las condiciones iniciales

$$u(x,0) = \sin(\pi x) \quad 0 \leq x \leq 1, \quad \text{y} \quad \frac{\partial}{\partial t}u(x,0) = 0, \quad 0 \leq x \leq 1,$$

usando para $n = 10$ y $m = 10$.

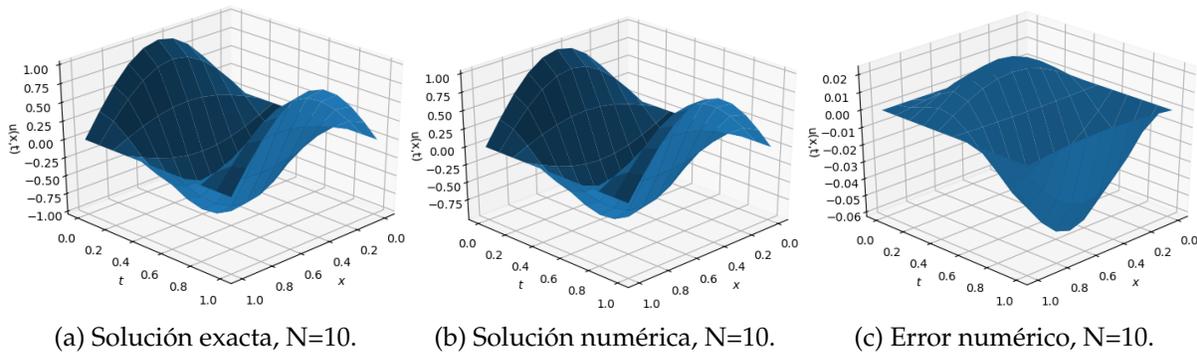


Figura 3.13: Cada gráfica muestra la solución exacta y numérica del respectivo problema y así como el error al compararse con la solución exacta $u(x,t) = \sin(\pi x)\cos(2\pi t)$ [37]

Problema 2. En la Fig. 3.14 se muestran los resultados al resolver el siguiente problema

$$\frac{\partial^2 u}{\partial t^2}(x, t) - \frac{\partial^2 u}{\partial x^2}(x, t) = 0, \quad 0 < x < 1, \quad 0 < t,$$

con las condiciones de límite

$$u(0, t) = u(1, t) = 0, \quad \text{para } 0 < t,$$

y condiciones iniciales

$$u(x, 0) = \sin(\pi x), \quad 0 \leq x \leq 1, \quad \text{y} \quad \frac{\partial u}{\partial t}(x, 0) = 0, \quad 0 \leq x \leq 1,$$

compara los resultados con la solución exacta

$$u(x, t) = \cos(\pi t)\sin(\pi x).$$

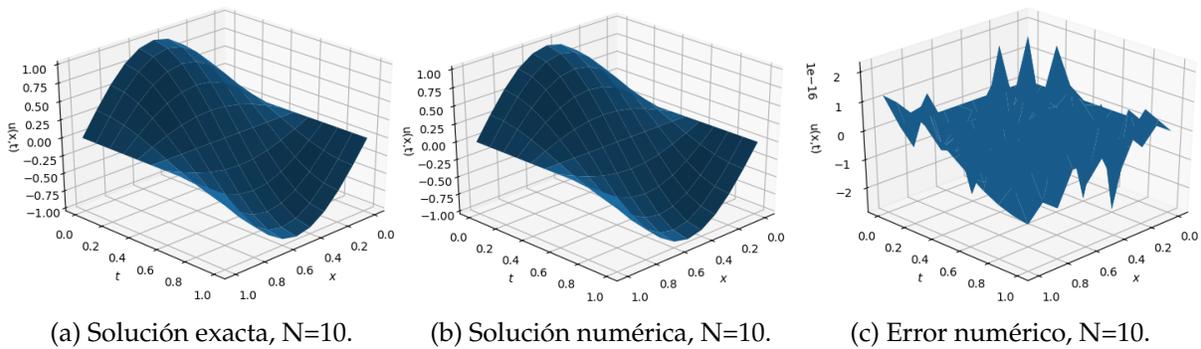


Figura 3.14: Cada gráfica muestra la solución exacta y numérica del respectivo problema.

En la subFig. 3.14c se muestra un gráfica por decirlo así “rara”, es decir, pareciera que algo no anda bien, sin embargo los resultados están bien pues algunos de ellos se calcularon de manera manual dando como resultado esos picos de la gráfica ya que se está graficando valores del orden $1e^{-16}$ por lo que estamos hablando de que el resultado es bastante exacto. Esto lo podemos notar respecto a las gráficas del error de los problemas 1 y 3, ya que el error es mucho más grande respecto a la gráfica del problema 2.

Problema 3. En la Fig. 3.15 se muestran los resultados al resolver el siguiente problema

$$\frac{\partial^2 u}{\partial t^2}(x, t) - \frac{1}{16\pi^2} \frac{\partial^2 u}{\partial x^2}(x, t) = 0, \quad 0 < x < 0.5, \quad 0 < t,$$

con las condiciones de límite

$$u(0, t) = u(0.5, t) = 0, \quad \text{para} \quad 0 < t,$$

y condiciones iniciales

$$u(x, 0) = 0, \quad 0 \leq x \leq 0.5, \quad \text{y} \quad \frac{\partial u}{\partial t}(x, 0) = \sin(4\pi x), \quad 0 \leq x \leq 0.5,$$

compara los resultados con la solución exacta

$$u(x, t) = \sin(t)\sin(4\pi x).$$

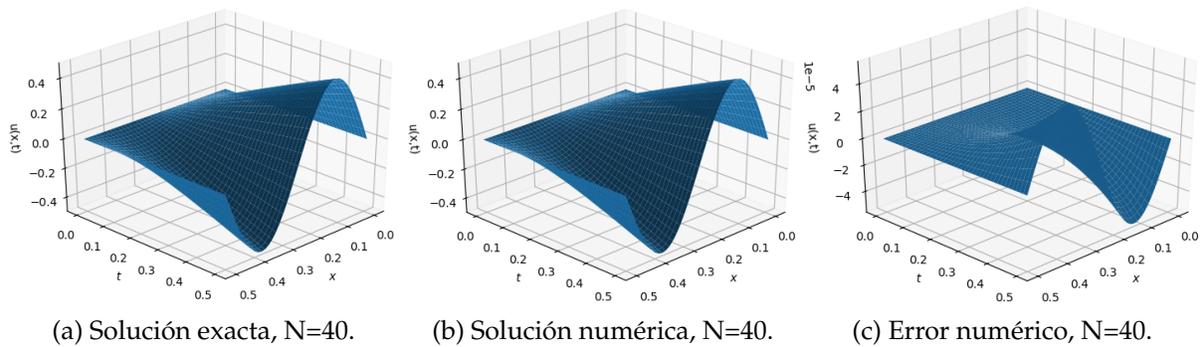


Figura 3.15: Cada gráfica muestra la solución exacta y numérica del respectivo problema.

Como se puede observar en esta ocasión se utilizó un tamaño de paso más fino, es decir, el dominio se dividió en $n = 40$ y $k = 40$ puntos.

3.3. Redes Neuronales Artificiales

Las RNAs [38] son un paradigma computacional, cuya estructura busca emular el aprendizaje como ocurre en las redes neuronales biológicas. El cerebro humano corresponde a un sistema altamente complejo, no lineal y paralelo. En términos sencillos lo anterior equivale a decir que puede realizar operaciones simultáneamente a diferencia de las computadoras comunes que son de tipo secuencial, es decir, realizan una operación a la vez. De esta manera las redes neuronales son capaces de realizar tareas como reconocimiento de patrones, regulación de energías eólicas y solares, además de aplicaciones en el área de la domótica¹ y recientemente han sorprendido en el ámbito de las artes visuales con imágenes generadas a partir de las redes neuronales de la compañía Google.

Las investigaciones realizadas en torno al estudio de RNAs, están motivadas en modelar la información del procesamiento de la información en sistemas nerviosos biológicos. Especialmente por la forma del funcionamiento del cerebro humano.

3.3.1. Forma de aprendizaje

Las primeras personas en introducir el concepto de redes neuronales fueron Warren McCulloch y Walter Pitts (1943) [39]. Ellos crearon un modelo informático para redes neuronales basado en las matemáticas y algoritmos denominados como lógica de umbral. Este modelo señaló el camino para que la investigación de redes neuronales se dividiera en dos enfoques distintos. El primero, centrado en los procesos biológicos en el cerebro y el otro en la aplicación de redes neuronales para la inteligencia artificial.

Antes de que McCulloch y Pitts introdujeran este concepto, en el año 1940 el psicólogo Donald Hebb creó una hipótesis de aprendizaje basado en el mecanismo de plasticidad neuronal que ahora se conoce como aprendizaje de Hebb [40]. Este tipo de aprendizaje se empezó a aplicar en los modelos computacionales con el uso de la máquina de Turing y posteriormente, las primeras máquinas computacionales en ser utilizadas fueron las calculadoras que simulaban una red o aprendizaje de Hebb.

¹Se llama *domótica* a los sistemas capaces de automatizar una vivienda o edificación de cualquier tipo, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado.

3.3.2. Redes neuronales como un modelo del cerebro humano

Desde que se empezó a estudiar la anatomía y estructura del tejido nervioso, los investigadores trataron de conocer la forma en cómo este tejido y los órganos que constituye, especialmente el cerebro que procesa la información que reciben de los órganos receptores, para dar una respuesta adecuada a sus estímulos. Aunque aún se está lejos de comprender completamente el funcionamiento y la estructura del sistema nervioso, se conoce con detalle la estructura de la neurona, como elemento básico del tejido nervioso y la forma cómo se estructura la corteza cerebral.

La neurona, como toda célula, consta de una membrana exterior conocida como cuerpo o soma, que la limita y le sirve de órgano de intercambio con el medio exterior. Este medio consta del citoplasma y el núcleo, el primero es el cuerpo principal de la célula donde radica el grueso de sus funciones y el segundo contiene el material genético de la célula, Fig. 3.16.

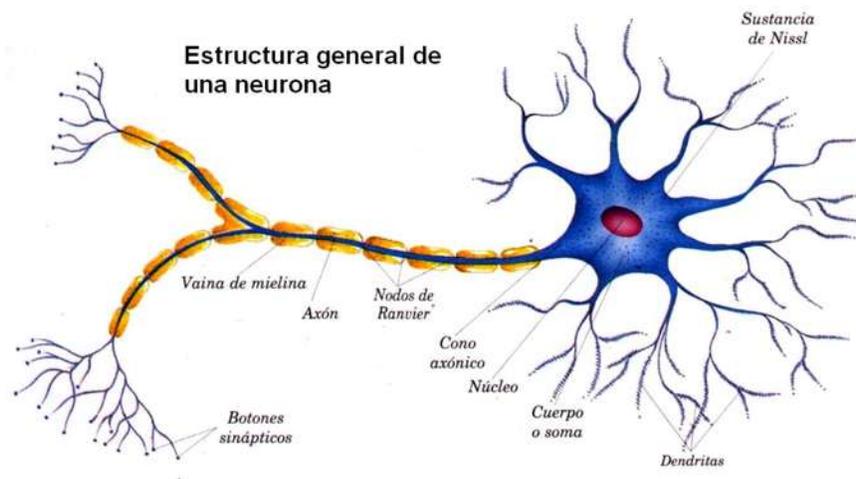


Figura 3.16: Esquema de una neurona biológica.

El citoplasma presenta unos alargamientos llamados dendritas, que son órganos de recepción. Ellos finalizan con un gran número de fibras que son conductores y éstas llevan la señal o impulso nervioso de los receptores hacia otras neuronas o incluso hacia la misma neurona. Estas fibras terminan en un pequeño corpúsculo llamado sinapsis, que constituye un relevador bioquímico y que sirve para transferir la señal de una neurona a otra.

Existen dos clases de sinapsis: actuadoras e inhibidoras. Las actuadoras son aquellas que favorecen el disparo de la neurona receptora y las inhibidoras dificultan este proceso.

Cuando se presenta un cierto desbalance entre las sinapsis actuadoras y las inhibidoras activas, la neurona dispara un impulso de salida, que constituye la respuesta de la neurona. Este impulso nervioso de salida es conducido por una prolongación cilíndrica

alargada (hasta de varios decímetros de largo) de la neurona, que se llama cilindro eje o **axón**, que en su extremo se divide en varias fibras para comunicarse con otras neuronas o con órganos efectores o motores como glándulas o músculos.

El citoplasma de las neuronas forma la masa gris de los centros nerviosos y el conjunto de cilindros ejes forma la masa blanca de aquéllos.

3.3.3. Topología de una red neuronal

Las redes neuronales artificiales tienen la forma sobre-simplificada de las neuronas biológicas aunque las artificiales son más sencillas, es decir, no posee tantos elementos a los ya mencionados.

En la parte superior de la Fig. 3.17 se muestra una neurona biológica y en la inferior una artificial. Se sabe que en las neuronas biológicas la entrada de información es recibida a través de las dendritas, después es enviada al núcleo y recibe una estimulación para que posteriormente pase a través del cuello del axón y finalmente obtener un resultado. De manera similar una neurona artificial, recibe entradas que son representadas como x_1, x_2, \dots, x_n y que son procesadas dentro de un cuerpo y por último se obtiene una o varias salidas.

En las neuronas biológicas, una vez que se procesó información de una neurona, ésta pasa por otra neurona y después por otra y así hasta obtener un resultado deseado, por lo que al final se forma una red neuronal biológica. La misma idea se aplica a la red artificial donde, si se hace un zoom en la parte del cuerpo lo que esta pasando son tres cosas:

1. Cuando entra una señal, se multiplica por un valor de peso asignado a esta entrada en particular. Es decir, si una neurona tiene tres entradas, entonces tiene tres pesos que se pueden ajustar individualmente.
2. Las señales de entrada modificadas se suman de un sólo valor.
3. Finalmente, el resultado del cálculo de la neurona se convierte en una señal de salida. Esto se hace alimentando el resultado a una función de activación (también llamada función de transferencia).

En la Fig. 3.18 se puede ver una red de neuronas artificiales formada por dos neuronas de entradas, una sola capa oculta² y una neurona de salida. Cada neurona artificial esta conectada por dos entradas y después por una salida.

²Las capas ocultas se refieren al conjunto de neuronas artificiales que se tenga, en el caso de la Fig. 3.18 se observa que se tiene una sola capa oculta con tres neuronas artificiales

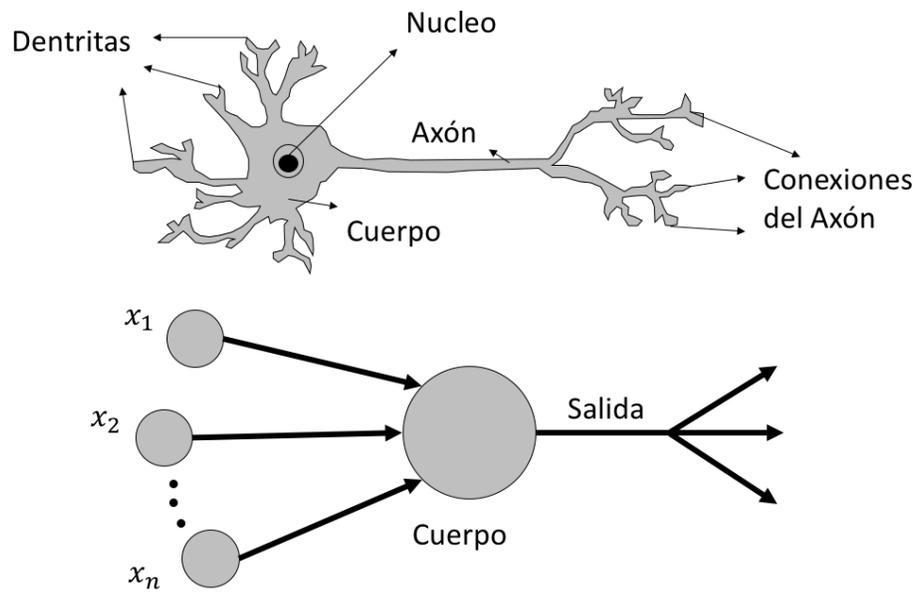


Figura 3.17: Esquema de una neurona artificial (abajo) comparada con una biológica (arriba).

Lo anterior, no significa que las redes neuronales artificiales estén restringidas, es decir, la red neuronal puede tener n entradas, k capas ocultas con l neuronas y m salidas.

En la Fig. 3.18 se observa que las entradas, las neuronas de la capa oculta y las salidas están representadas con círculos, además están dirigidas por una flecha que representa los pesos w de la red neuronal.

El algoritmo de entrenamiento retropropagación, surge el caso especial en el que las redes tienen más de una capa oculta. En esta sección se explicará y ejemplificará la construcción de una red neuronal utilizando el algoritmo de aprendizaje. Para esto se pretende dar un conjunto de datos para que sean propagados a través de la red neuronal, hacer una comparación de resultados respecto a la clasificación obtenida, posteriormente hacer un respectivo ajuste de peso y luego repetir con otro conjunto de datos, además se darán fórmulas explícitas para actualizar los pesos y mostrar cómo pueden ser calculados usando operaciones algebraicas lineales.

3.3.4. Red neuronal generalizada

En la sección 3.3.3 se habló tanto de las redes neuronales biológicas como las redes neuronales artificiales y en particular de un perceptrón. En la Fig. 3.19, podemos ver la estructura general de una red neuronal artificial.

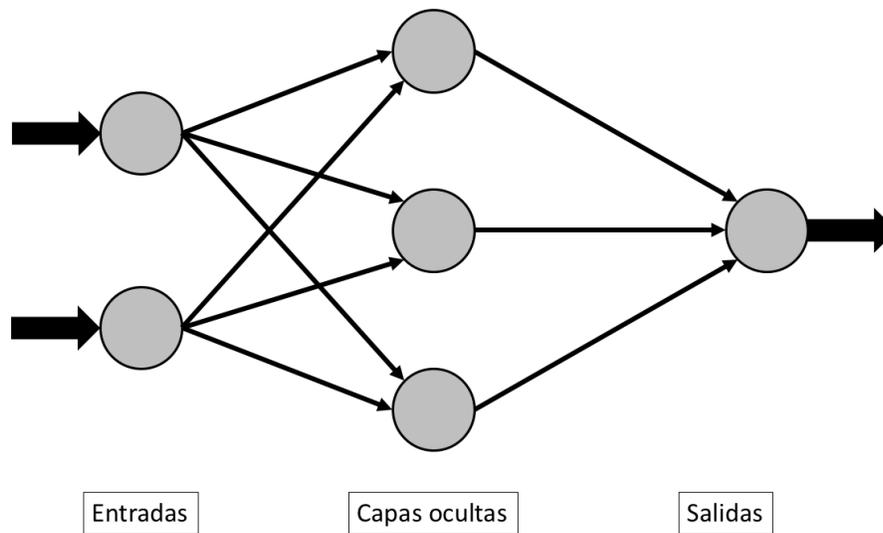


Figura 3.18: Esquema de una neurona artificial.

Dentro del aprendizaje máquina, la retropropagación es un algoritmo de aprendizaje supervisado o de clasificación, básicamente se puede ver como una función de activación que decide si un conjunto de datos de entrada pertenece a una clase específica o no. Una de las cosas interesantes de este algoritmo es que permite un aprendizaje en tiempo real ya que el conjunto de elementos son entrenados un paso a la vez, lo que permite actualizar valores de entrada como sea requerido.

Cuando se combinan múltiples neuronas, cada neurona de salida funciona independientemente de todas las demás por lo tanto, aprender cada producto se puede considerar de forma aislada.

Para simplificar un poco más las cosas y tener un mejor entendimiento de la red neuronal artificial, se tomará $k = 1$, esto es una sola capa oculta como se muestra en la Fig. 3.20, en la que se tienen l neuronas en la capa oculta, n entradas, m salidas y dos conjuntos de datos que corresponde a los pesos W^n $n = 1, 2, \dots$, en particular W^1 que se ubica entre las entradas y las neuronas de la capa oculta y W^2 que está entre las neuronas de la capa oculta y las salidas.

Se tiene un conjunto D de entradas con n elementos para esta red neuronal. Las entradas son datos de información que serán propagados en dicha red. Se puede representar a este conjunto de datos como un arreglo, tal como

$$D = x_i = x_1, x_2, \dots, x_n, \quad (3.22)$$

como en el caso del perceptrón, en esta red neuronal los pesos W^1 son números generados aleatoriamente, por lo general entre 1 y -1. Para fines de este trabajo se elegirá una cantidad

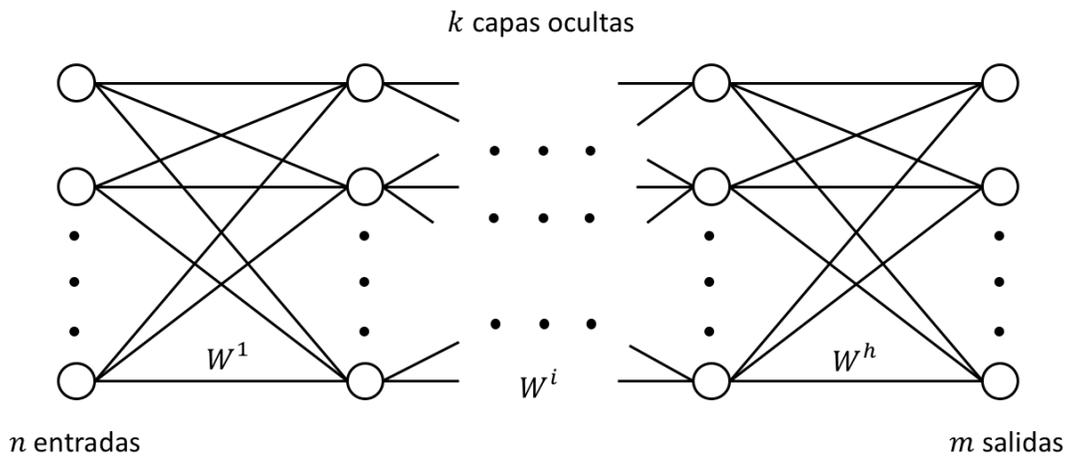


Figura 3.19: Estructura de una red neuronal generalizada, con n entradas, k capas ocultas y m salidas, se muestran también los pesos de las neuronas representados por W^h .

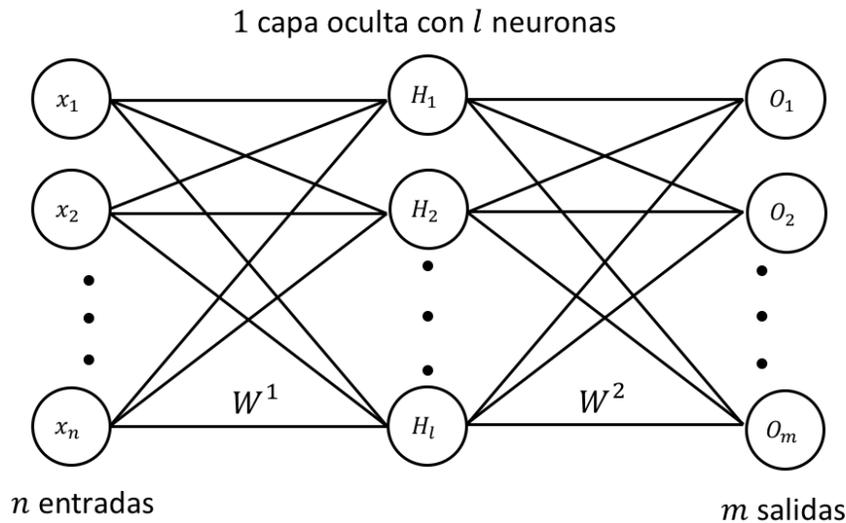


Figura 3.20: Estructura de una red neuronal, con n entradas, $k = 1$ capas ocultas con l neuronas y m salidas, además de sus respectivos pesos W^1 y W^2 .

l neuronas. Para este caso ($k = 1$), a partir de ahora la notación correspondiente a los pesos W^1 se escribirán como w_{nl} .

Si se elige a una sola neurona de la capa oculta H_1 , se regresaría al caso del perceptrón, lo que significa que las dimensiones de la matriz serían del tamaño w_{n1} y se resolvería la Ec. (3.23) para $l = 1$, lo mismo sería para $l = 2$ con los pesos correspondientes w_{n2} con

neurona H_2 , al igual que para $l = 3$ con neurona H_3 y así sucesivamente hasta llegar a la neurona H_l . Es claro tener en cuenta que ahora H es un arreglo de datos con l conjunto de elementos.

$$\begin{aligned} H_1 &= \sum_{i=1}^n x_i w_{i1} \\ H_2 &= \sum_{i=1}^n x_i w_{i2} \\ &\vdots \\ H_l &= \sum_{i=1}^n x_i w_{il} \end{aligned} \quad (3.23)$$

Hasta ahora la Ec. (3.23) calcula el valor de cada neurona l , entonces como en el caso del perceptrón, los valores de cada neurona se evalúan en una función de activación y esta prácticamente le dice al algoritmo que camino tomar.

Para este tipo de red neuronal que se está analizando existen una gran cantidad de funciones de activación.

Una de las funciones de activación más popular es la función sigmoidea, ésta es una función real que se define como $f_c : \mathbb{R} \rightarrow (0, 1)$ con la expresión

$$f_c = \frac{1}{1 + e^{-cx}} \quad (3.24)$$

con c contante que puede ser seccionada de manera arbitraria.

En la Fig. 3.21 se puede ver la función sigmoidea y la manera en la que cambia de acuerdo a como varía el valor constante c . Como se puede observar los valores que se eligieron son para $c = 1$, $c = 2$, $c = 3$. Además un caso particular de la función sigmoidea es cuando el valor c es bastante grande, es decir, cuando $c \rightarrow \infty$ la función sigmoidea converge a la función escalón.

A fin de simplificar todos los valores de c , para caracterizar expresiones futuras para $c = 1, 2, 3, \dots$ lo que se hace es tomar a $c = 1$. Entonces la notación de la Ec. (3.24) para $f_1(x)$ será $f(x)$.

Otra de la funciones de activación es la función rampa, la cual está definida como

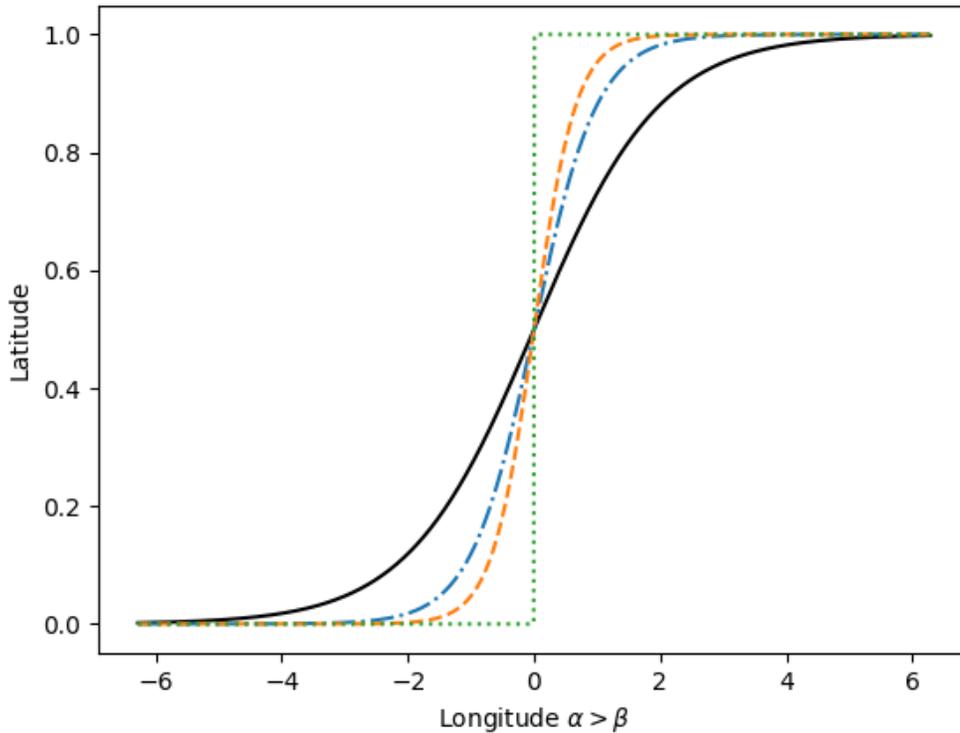


Figura 3.21: Se muestran cuatro funciones sigmoidea, la función con la línea negra uniforme es cuando $c = 1$, la línea “· - ·” de color azul es cuando $c = 2$, la línea “- - -” de color naranja es cuando $c = 3$ y la línea “· · ·” de color verde es cuando $c \rightarrow \infty$.

$$ramp(x) = \begin{cases} -1 & \text{si } x \leq -1, \\ x & \text{si } -1 < x < 1, \\ 1 & \text{si } 1 \geq x. \end{cases} \quad (3.25)$$

La función tangente hiperbólica es otra de las funciones de activación. Esta es una función real con dominio en $[-\infty, \infty]$ e imagen en $(-1, 1)$. Además esta función se puede expresar de varias maneras pero siendo la más común

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.26)$$

Aunque existen muchas otras más funciones de activación, las ecuaciones (3.24), (3.25) y (3.26), son de las más populares que se utilizan en las redes neuronales artificiales. Estas son funciones de activación continuas, es decir, el límite existe en todo punto de la fun-

ción a diferencia de la función de activación que se utiliza para el perceptrón, que es una función de activación discontinua en el punto cero.

Retomando la Ec. (3.23), el siguiente paso es evaluar el valor H_l dentro de la función de activación, como se hizo en el perceptrón. Entonces, sea la función de activación que se haya elegido quedaría de la forma

$$H_l = F \left(\sum_{i=1}^n x_i w_{il} \right). \quad (3.27)$$

Es necesario calcular el valor de todas las salidas O_m , la forma de hacerlo es haciendo operaciones, como las que se hicieron para calcular los H_l . En la Fig. 3.20, del lado derecho de la imagen se ve la conexión entre las l neuronas y las m salidas que éstas se encuentran conectadas por los pesos W^2 , que a partir de ahora se escribirán como \tilde{w}_{lm} . El valor de los pesos \tilde{w}_{lm} son generados de manera aleatoria entre 1 y -1. Como en la Ec. (3.27) ya se calculó el valor de las H_l neuronas, ahora para encontrar el valor de las salidas, simplemente se sigue propagando hacia adelante y esta vez utilizando a los pesos \tilde{w}_{lm} correspondientes, así que

$$O_m = \sum_{l=1}^n H_l \tilde{w}_{lm}, \quad (3.28)$$

y evaluando dentro de la función de activación, quedaría

$$O_m = F \left(\sum_{l=1}^n H_l \tilde{w}_{lm} \right). \quad (3.29)$$

Nuevamente O es un arreglo de datos con m conjuntos de elementos que representan las salidas.

Se ha finalizado la descripción del funcionamiento de la propagación hacia adelante y se han generado O_m salidas. Ahora se introduce un valor deseado T para cada salida lo que significa que se tiene un número de m valores deseados, o sea T_m . Como en el caso del perceptrón se generó una sola salida y esta generó un valor de salida igual a **uno** debido a la función de activación. Ahora supongamos que el problema requiere que la salida sea igual a **cero** (esto representa el valor deseado), por lo tanto, designa un error generado en la salida y es necesario corregirlo.

Al suponer que se cuentan con salidas, por ejemplo, si se tienen 3 salidas esto implica que se tiene 3 valores deseados para cada ejemplo de esta red neuronal. Ahora si se cuenta con un número m de salidas y m valores deseados respectivamente, por ende, surge un error de cada neurona y se calcula de la siguiente manera

$$E_m = \frac{1}{2} \|T_m - O_m\|^2. \tag{3.30}$$

La Ec. (3.30) representa el error local de cada l neurona. Entonces una vez calculado la función de error de cada neurona, lo que procede ahora es sumar todos los errores, es decir, recoger todos los errores cuadráticos tal como

$$E^G = \sum_{i=1}^m E_i \tag{3.31}$$

En la Fig. 3.22 del lado derecho se muestra cómo se calculan las ecuaciones (3.30) y (3.31) que representa dichos errores tanto local como global, respectivamente. Esto después de haber calculado la Ec. (3.29).

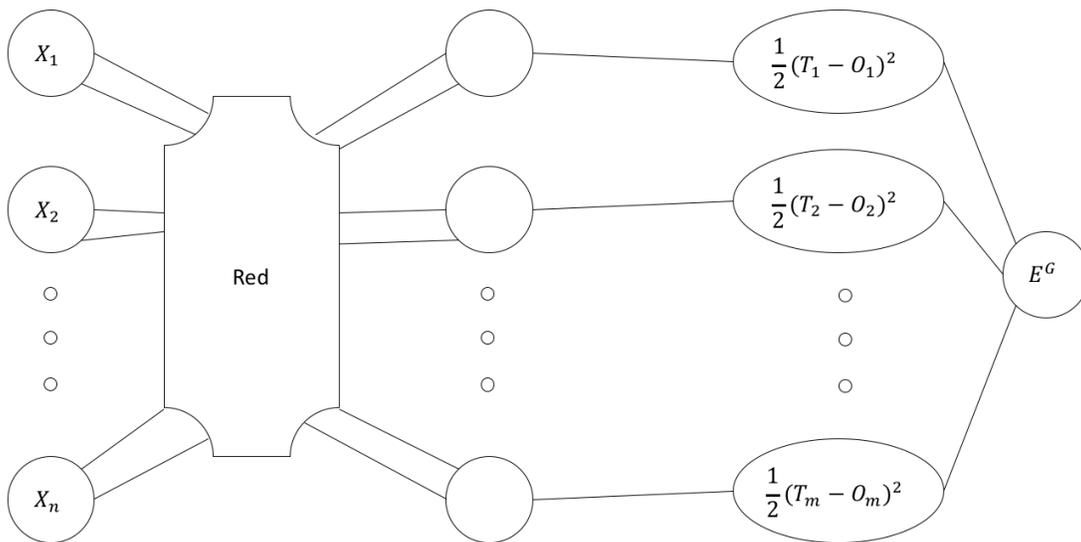


Figura 3.22: Red neuronal donde se muestra el cálculo del error local y global E^G , donde O_m representa las salidas y T_m representa los valores esperados.

Hasta ahora se ha hecho la mitad de algoritmo de entrenamiento. El cálculo está hecho para $k = 1$ capas ocultas con el fin de simplificar las cuentas. Pero, si se tuvieran k capas

ocultas, la Ec. (3.27) cambiaría un poco la notación, es decir, para $k = 1$ con l_1 neuronas se escribe como $H_{l_1}^1$, para $k = 2$ con l_2 neuronas se tendría $H_{l_2}^2$ y así sucesivamente hasta las k capas ocultas. En general se remplazaría dicha ecuación por $H_{l_q}^k$ con l_q neuronas. La cantidad q de cada l_q neuronas pueden o no ser iguales, dicho de otro modo, $l_1 = 10$ neuronas, $l_2 = 50$ neuronas y así de manera consecutiva, no hay alguna restricción. Sin embargo, una vez definido la cantidad de capas ocultas y de neuronas, implica que se ha construido una única estructura de una red neuronal, y si se cambian los valores de k capas ocultas o l neuronas, denota otra única estructura de una red neuronal, lo cual tiene una consecuencia al momento de mostrar resultado, en otros términos, los resultados pueden mejorar o empeorar.

Se ha calculado la *propagación hacia adelante* que genera salidas O_m y además se ha calculado la *función error* de cada salida. Además del error global de la red neuronal artificial, el siguiente paso es que la red neuronal sea capaz de “aprender” un conjunto de datos.

Capítulo 4

Aplicación a la propagación de mosquitos

Los mosquitos representan un problema de salud pública importante ya que son responsables de más 750,000 muertes al año frente a las 50,000 muertes provocadas por las serpientes o las apenas 10 personas que matan los tiburones [43] [44] [45].

En este capítulo mostraremos los resultados de los capítulos descritos anteriormente, primero comenzaremos con mostrar como resolver la ecuación de reacción difusión con los métodos aprendidos en el capítulo *Métodos numéricos*. Posteriormente, la construcción del dispositivo electrónico que medirá la humedad y temperatura, además atraerá y contendrá a los mosquitos para hacer su conteo de datos. Además en esta parte se probaron distintos métodos de atracción para determinar cual es mejor.

4.1. Ecuación de reacción difusión

Las ecuaciones diferenciales son usadas para modelar problemas tanto científicos como de ingeniería que involucran cambios de alguna variable respecto a otra. Muchos de esos problemas requieren solución de un problema de valor inicial, que es la solución a la ecuación diferencial que satisface una condición inicial dada.

Sea la ecuación de reacción difusión dada por

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + R(u), \quad (4.1)$$

donde $u(x, t)$ representa una función vectorial desconocida, D representa el coeficiente de difusión y R el coeficiente de reacción. Estos coeficientes están relacionados entre sí,

es decir, basta con cambiarlos para estudiar diferentes sistemas de comportamiento. Si eliminamos el término de reacción $R(u)$ entonces esto representa un proceso de difusión, siendo correspondiente a la ecuación de calor.

Para resolver esta ecuación diferencial parcial de segundo orden no lineal nuevamente resolvemos mediante el método de diferencias finitas. Para esta parte se utilizará el método de diferencias centradas para la parte espacial y diferencia hacia adelante para la parte temporal.

Para la parte espacial queda de la siguiente manera donde se parte por expandir a la variable dependiente u para el tercer término del polinomio de Taylor primero para el punto x_i evaluado en el punto x_{i-1} y x_{i+1} , además se asume que $u \in C^4[x_{i-1}, x_{i+1}]$,

$$\begin{aligned} u(x_{i-1}, t_j) &= u(x_i - h, t_j) \\ &= u(x_i, t_j) - hu'(x_i, t_j) + \frac{h^2}{2}u''(x_i, t_j) - \frac{h^3}{6}u'''(x_i, t_j) + \frac{h^4}{24}u^{(4)}(\xi_i^-, t_j), \end{aligned}$$

para algún ξ_i^- dentro de (x_{i-1}, x_i) y

$$\begin{aligned} u(x_{i+1}, t_j) &= u(x_i + h, t_j) \\ &= u(x_i, t_j) + hu'(x_i, t_j) + \frac{h^2}{2}u''(x_i, t_j) + \frac{h^3}{6}u'''(x_i, t_j) + \frac{h^4}{24}u^{(4)}(\xi_i^+, t_j), \end{aligned}$$

para algún ξ_i^+ dentro de (x_i, x_{i+1}) . Ahora sumamos esas dos ecuaciones y resolviendo aritméticamente tenemos

$$u(x_{i-1}, t_j) + u(x_{i+1}, t_j) = 2u(x_i, t_j) + h^2u''(x_i, t_j) + \frac{h^4}{24}[u^{(4)}(\xi_i^-, t_j) + u^{(4)}(\xi_i^+, t_j)]$$

y resolviendo para $u''(x_i, t_j)$ resulta

$$u''(x_i, t_j) = \frac{1}{h^2}[u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)] - \frac{h^4}{24}[u^{(4)}(\xi_i^-, t_j) + u^{(4)}(\xi_i^+, t_j)].$$

Este es un resultado bastante bueno, debido a que se puede simplificar con el teorema del valor medio el cual nos simplificará el término del error, dando

$$u''(x_i, t_j) = \frac{1}{h^2}[u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)] - \frac{h^2}{12}u^{(4)}(\xi_i, t_j), \quad (4.2)$$

donde $\xi_i \in (x_{i-1}, x_{i+1})$.

De una manera similar para la variable temporal t_j evaluado en $u \in C^2[t_j, x_{j+1}]$,

$$\begin{aligned} u(x_i, t_{j+1}) &= u(x_i, t_j + k) \\ &= u(x_i, t_j) - ku'(x_i, t_j) + \frac{k^2}{2}u''(x_i, \eta_j^+), \end{aligned}$$

para algún η_j^+ dentro de (t_j, t_{j+1}) . Resolviendo para $u'(x_i, t_j)$ resulta

$$u'(x_i, t_j) = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} + \frac{k}{2} u''(x_i, \eta_j^+) \quad (4.3)$$

Ahora sustituyendo las ecuaciones (4.2) y (4.3) en la Ec (4.1) se tiene entonces que

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} = D \frac{u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)}{h^2} + R(u(x_i, t_j)). \quad (4.4)$$

Entonces el método de diferenciación donde $w_{i,j}$ aproxima a $u(x_i, t_j)$ de las ecuaciones las ecuaciones (4.2) y (4.3) es

$$\frac{w_{i,j+1} - w_{ij}}{k} - D \frac{w_{i,j+1} - 2w_{i,j} + w_{i-1,j}}{h^2} - R(w_{i,j}) = 0. \quad (4.5)$$

El error local de truncamineto es

$$\tau_{ij} = \frac{k}{2} \frac{\partial^2}{\partial t^2} u(x_i, \eta_j) - D \frac{h^2}{12} \frac{\partial^4}{\partial x^4} u(\xi_i, t_j). \quad (4.6)$$

Finalmente resolviendo a la Ec (4.5) para $w_{i,j+1}$ da

$$w_{i,j+1} = \left(1 - \frac{2Dk}{k^2}\right) w_{ij} + D \frac{k}{h^2} (w_{i+1,j} + w_{i-1,j}) + R(w_{ij}), \quad (4.7)$$

para cada $i = 1, 2, \dots, m-1$ y $j = 1, 2, \dots$. Por lo que

$$w_{i0} = f(x_i).$$

Problema 1. En la Fig. 4.1 se muestran los resultados al resolver el siguiente problema

$$\frac{\partial^2}{\partial t} u(x, t) - \frac{\partial^2}{\partial x^2} u(x, t) = 0, \quad 0 < x < 1, \quad 0 < t,$$

con las condiciones de frontera

$$u(0, t) = u(1, t) = 0, \quad \text{para } 0 < t,$$

y las condiciones iniciales

$$u(x, 0) = \sin(\pi x) \quad 0 \leq x \leq 1,$$

usando para $n = 10$ y $m = 2000$.

Comparando el resultado en $t = 0.5$ con la solución exacta

$$u(x, t) = e^{-\pi^2 t} \sin(\pi x).$$

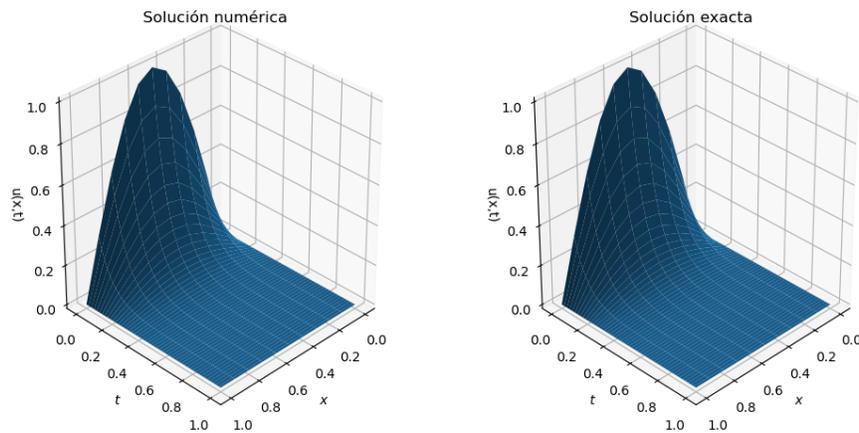


Figura 4.1: Solución numérica correspondiente al problema 1 de esta sección.

En la Fig. 4.1 como podemos observar la condición inicial esta dada por un $\sin(\pi x)$, tal y como nos indica las condiciones iniciales $u(x, 0) = \sin(\pi x)$, es decir, cuando $t = 0$ en todo x vale la función seno.

Desde este punto podemos prácticamente ya trabajar y tomar el resultado de la Fig. 4.1 como la propagación de mosquitos, donde la parte central y más alta de la función seno lo consideramos como la mayor concentración de mosquitos y mientras esta se va alejando va evolucionando en el tiempo esta función se va difundiendo a lo largo de todo el dominio.

En la Fig. 4.2 tenemos la forma de representar la ubicación de los detectores electrónicos, en este caso lo representamos mediante puntos rojos. Como se puede observar es evidente que en el detector 1 se obtenga un conteo de datos mayor que el de detector número 2 debido a su posición.

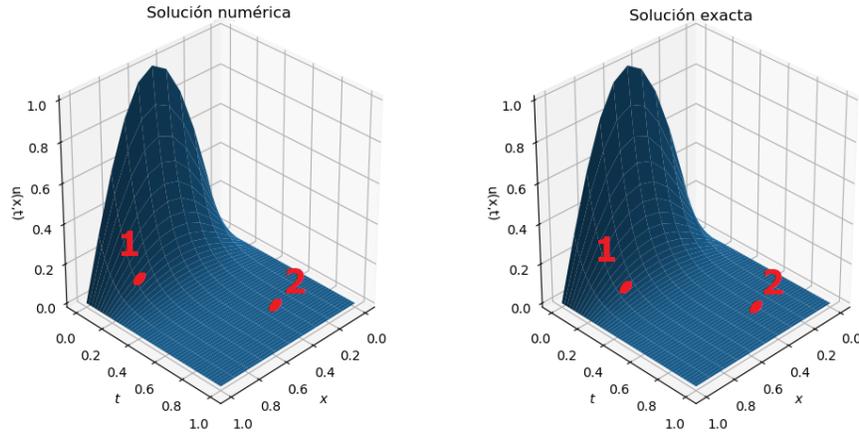


Figura 4.2: Solución numérica correspondiente al problema 1 de esta sección con los correspondientes detectores electrónicos.

Para esta misma idea es conveniente que se utilice una ecuación que se adecue más al problema que estamos resolviendo, siendo la ecuación famosa conocida como el sombrero mexicano o mejor dicho una función gaussiana la cual se remplazará la condición inicial del perfil senoidal trabajado en el problema 1, esta ecuación esta dada por

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}, \tag{4.8}$$

con a, b y c como constantes arbitrarias. Comúnmente $a = \frac{1}{\sigma\sqrt{2\pi}}$.

Si elegimos ciertos valores como se muestra en al Fig. 4.3 obtenemos diferentes formas de la función.

Para el problema 1 de esta misma sección sólo que cambiando la condición inicial para la posición por una función gaussiana en una dimensión. Tendríamos el resultado mostrado en la Fig. 4.4 del lado izquierdo mientras que del lado derecho mostramos una representación con los dispositivos electrónicos. Respecto al conteo de la distribución de mosquitos se obtiene resultados similares a los dichos anteriormente; es decir, del dispositivo electrónico 1 se obtendría más datos que del dispositivo electrónico 2. Sólo basta con poner más dispositivos electrónicos para obtener mucha más información.

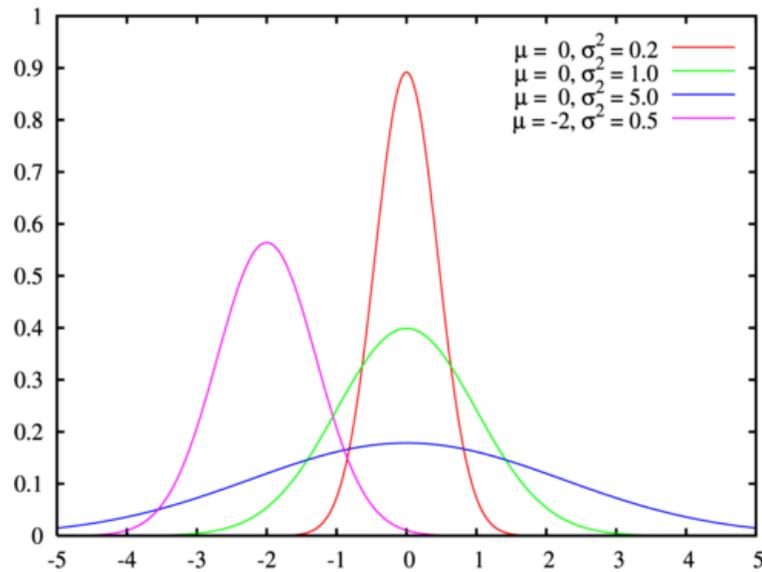


Figura 4.3: Función gaussiana donde $a = \frac{1}{\sigma\sqrt{2\pi}}$, $b = \mu$, $c = \sigma$.

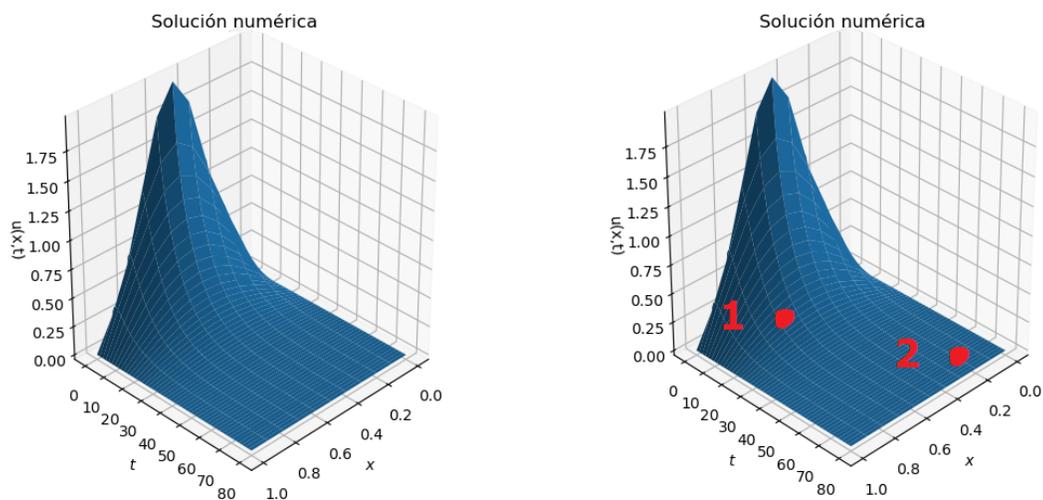


Figura 4.4: Función gaussiana y dispositivos electrónicos.

4.2. Construcción del dispositivo electrónico

Tal y como se describió en el capítulo 2, ahora es momento de construir dicho dispositivo electrónico que sea capaz de medir variables climatológicas descritas.

Inicialmente la construcción del dispositivo electrónico consistió de varias etapas. Al ser un dispositivo tal que debemos de estar probando la ubicación de sensores, atracción de mosquitos, captura de mosquitos y por último la ubicación del dispositivo electrónico.



Figura 4.5: Se colocó el que cuenta con la atracción de CO_2 .

4.2.1. Etapa uno

La primera etapa consistió en probar los métodos de atracción de mosquitos. Uno de ellos está basado en la atracción mediante el CO_2 y el otro método consiste en la atracción con luz ultravioleta.

Construir una trampa para mosquitos suele no ser tan fácil ya que uno se enfrenta a problemas de como la forma en atraerlos y luego a como capturarlos. Básicamente, para el problema de la atracción, una forma de hacerlo es mediante CO_2 ya que es lo que emiten los seres vivos al exhalar por la nariz o boca una vez inhalado oxígeno, esta forma de atracción nos permite implementar la primera opción. En la Fig. 4.5 se muestra una botella completamente asilada con cinta negra resistente al calor, la cual utilizan los electricistas para enrollar cables pelados. La forma en la que se genera el CO_2 es mediante 3 ingredientes: agua caliente sin cloro la cual al hervirla por 7 minutos el cloro se elimina, levadura y azúcar. El proceso de generar CO_2 mediante estos ingredientes es simplemente mezclarlos. Una forma de comprobar que si se genera CO_2 es colocarla en una botella de plástico, cerrarla bien y posteriormente agitarla después de unos minutos la botella se empezará a inflar.

4.2.2. Etapa dos

El primer paso fue elegir el tipo de material a utilizar, desde un inicio se pensó en utilizar plástico proporcionado por la impresora 3D. Previamente antes de construirlo a través de la impresora 3D se pusieron a prueba distintas formas.



Figura 4.6: Se colocó el dispositivo electrónico en un jardín con toma corriente de manera ilimitada.

La estructura del dispositivo electrónico consiste en un material muy conocido y barato: el MDF. Productos derivados de madera son una propuesta interesante en la ampliación de los materiales que son empleados en la industria de la construcción civil, de muebles y otros usos. Entre los productos derivados de madera se destacan los tableros de fibra de media densidad (Medium Density Fiberboard - MDF), un producto que ofrece buena condición de trabajo. MDF es fabricado en seco, hecho con fibras lignocelulósicas, combinadas con resina, compactados por prensado en caliente. Este trabajo presenta la producción de MDF utilizando fibras de eucalipto y diferentes porcentajes de adhesivo poliuretano de origen natural. Después de la fabricación fueron realizados ensayos de caracterizaciones físico-mecánicas y comparados con especificaciones contenidas en la norma Europea de tableros MDF (Euro MDF Board) [46].

Este material es un producto bastante bueno para ser manipulado tanto para herramientas eléctricas como manuales, incluso el lijado a mano requiere de poco esfuerzo, además es un producto muy resistente. En el mercado existen diferentes dimensiones del producto en cuestión al grosor hay desde 2.7 mm hasta los 30 mm. Para este proyecto se utilizó un MDF de grosor de 12 mm.

4.2.3. Etapa tres

La colocación de los módulos electrónicos suele ser un poco fácil en cuestión de elegir la ubicación, ya que una de las ventajas es que estos módulos son pequeños y ocupan poco espacio.

4.2.4. Validación

El método de atracción que se utilizó en la etapa uno es mediante el uso de CO_2 . Este método es bastante bueno para atraer a múltiples insectos desde moscas, abejas, mosquitos, polomillas, entre otros insectos, incluso al haber agua con nutrientes esto permite a ciertos insectos poner sus huevos para poder reproducirse tal y como se ve en la Fig. 4.7. Los gusanos de color blanco amarillento, son pequeñas lombrices que aparecen en la imagen. Estos gusanos se parecen mucho a los que aparecen cuando se deja desperdicio de comida en el bote de basura y pasa un tiempo. Normalmente los gusanos blancos aparecen en la cocina, debajo de las bolsas de basura y cerca o debajo de las alfombras. La razón por la cual aparecen es porque las moscas pusieron huevos en esa zona. Estos gusanos son las larvas de las moscas. El elemento que los atrae principal es el olor a comida en descomposición. Por esta razón es que no se decide por optar por este método ya que se vuelve un poco complicado hacerle mantenimiento; además a la larga esto se convierte en factor costo en comparación al método de atracción de luz ultravioleta.



Figura 4.7: Gusanos que aparecen en una botella utilizando el método de atracción de CO_2 .

Por otro lado, el sistema que se desarrolló en la etapa dos es utilizando la luz ultravioleta, en cuestiones de mantenimiento sólo se requiere cuando algún LED se haya quemado o terminado su tiempo de vida. Los LEDs generalmente tienen un tiempo de vida de 500,000 horas siempre y cuando se respete el voltaje soportado por cada LED. Este método en particular es el que se decidió utilizar por su simpleza y manejabilidad. Es un sistema bastante ligero y de dimensiones medianas. Este tiene una altura de (18 x 24 x 9) en centímetros.

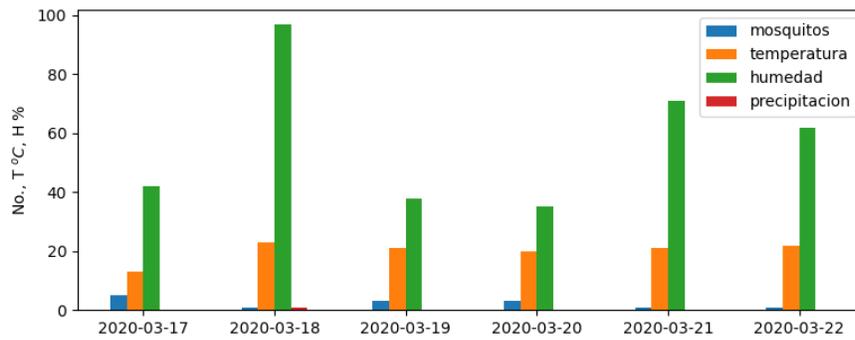
4.3. Gráfica de datos

El dispositivo electrónico construido es capaz de hacer mediciones tanto de la temperatura ($^{\circ}\text{C}$) como de la humedad (%) proporcionadas por el módulo DHT22 además de la captura de mosquitos. En las gráficas de la Fig. 4.8 a la 4.29 se muestran los datos registrados en forma de histograma donde el eje de las Y's representan el conjunto de datos medidos de la temperatura, humedad, precipitación y la cantidad de mosquitos. Además en cada Fig. en la parte de abajo de la gráfica de histograma se muestra una tabla con datos medidos de cada día.

La cantidad de mosquitos para este dispositivo en promedio diariamente se capturaba 2.7 mosquitos. Aquí es un poco difícil determinar estas cifras debido a que no se sabía cuantos mosquitos había o si ya no los había además la eficiencia del dispositivo también es difícil de calcular ya que no se tiene otro dispositivo con el que comparar.

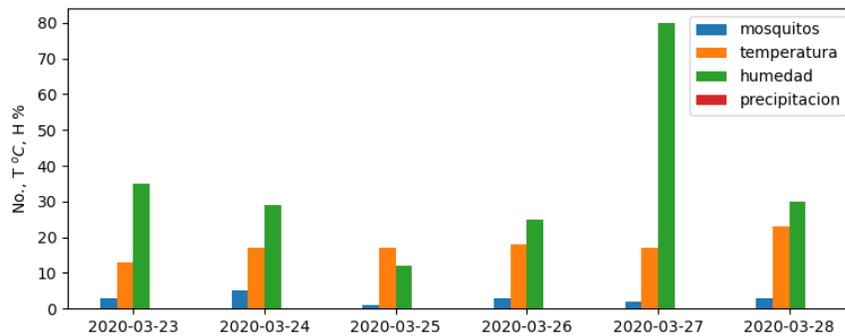
El dispositivo electrónico se puso a trabajar durante dos horas al día (aprox. 19:00 a las 22:00 horas) por 131 días el cual comenzó desde el 17 de Marzo de 2020 hasta el 26 de Julio de 2020. Los datos obtenidos de la temperatura oscilaban entre los 13°C hasta 26°C debido al cambio climático se tiene distintas variaciones. En este caso el valor de la temperatura es el promedio de los datos de la medición de cada 10 minutos por lo que obtenía todos los días una cantidad de 12 datos. la temperatura de cada día oscilaba alrededor de 1.2°C en promedio durante las dos horas cuando se ponía el dispositivo a trabajar.

De manera similar a la forma en que la temperatura se midió y se recolectaron los datos respecto a humedad, aquí se obtuvieron 12 datos los cuales se miden en un porcentaje de 0 a 100 % donde 0% significa que hay poca humedad y 100 % humedad alta en el ambiente. Un dato observado es que cuando llovía antes de la hora de medición había humedad alta y cuando llovía en el momento y la humedad estaba algo baja, se esperaba que aumentara de manera considerable la humedad pero no siempre sucedía así. Otro dato relacionado en algunos datos registrados es que la temperatura aumenta cuando llueve esto se debe a una propiedad del agua: *elevado calor de vaporización*. Por término medio el número de enlaces por puente de hidrógeno, con una energía de enlace de 2.8 Kcal/mol, es 4 para el agua sólida, 3.5 para la líquida y 0 para la gaseosa. Ello significa que su calor de ebullición equivale a 80 cal/g y que el de vaporización es de unas 540 cal/g. Este valor elevado permite eliminar el exceso de calor de cantidades relativamente pequeñas de agua. Cabe mencionar que los datos para precipitación fueron medidos de manera manual y se registraron sólo si llovía en ese momento en el que el dispositivo estaba en funcionamiento, a partir del primer día se colocó la caja de 7 pm a 9 pm. Una humedad sobre el 80 % era un indicativo de que estaba lloviendo y previamente a esas horas también había llovido. Sin embargo hubo días donde la humedad estaba sobre el 80 % y no llovía en ese momento pero si había llovido previamente.



	Día 0	Día 1	Día 2	Día 3	Día 4	Día 5
Mosquitos	5.0	1.0	3.0	3.0	1.0	1.0
Temperatura	13.0	23.0	21.0	20.0	21.0	22.0
Humedad	42.0	97.0	38.0	35.0	71.0	62.0
Precipitación	0.0	1.0	0.0	0.0	0.0	0.0

Figura 4.8: Datos correspondientes a la fecha medida de 17 de Marzo al 22 de Marzo.



	Día 6	Día 7	Día 8	Día 9	Día 10	Día 11
Mosquitos	3.0	5.0	1.0	3.0	2.0	3.0
Temperatura	13.0	17.0	17.0	18.0	17.0	23.0
Humedad	35.0	29.0	12.0	25.0	80.0	30.0
Precipitación	0.0	0.0	0.0	0.0	0.0	0.0

Figura 4.9: Datos correspondientes a la fecha medida de 23 de Marzo al 28 de Marzo.

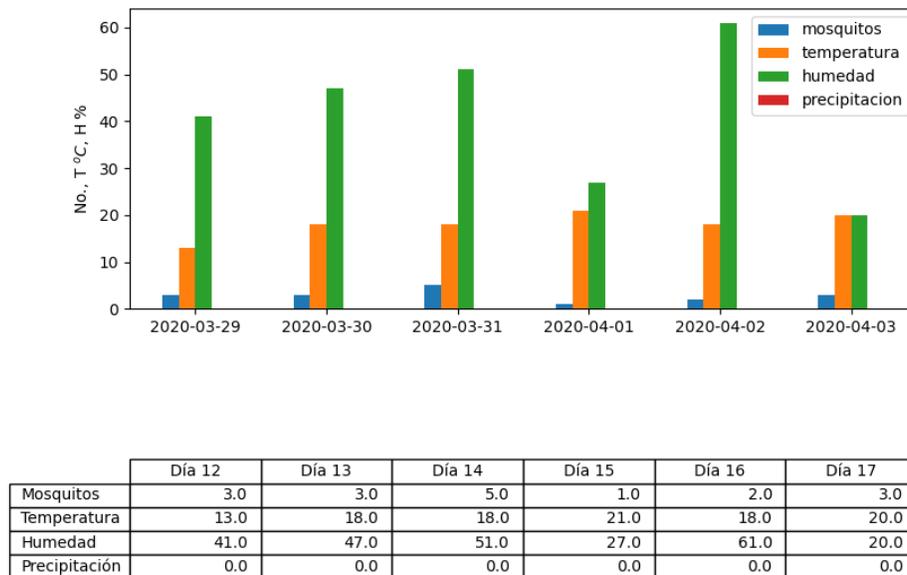


Figura 4.10: Datos correspondientes a la fecha medida de 29 de Marzo al 3 de Abril.

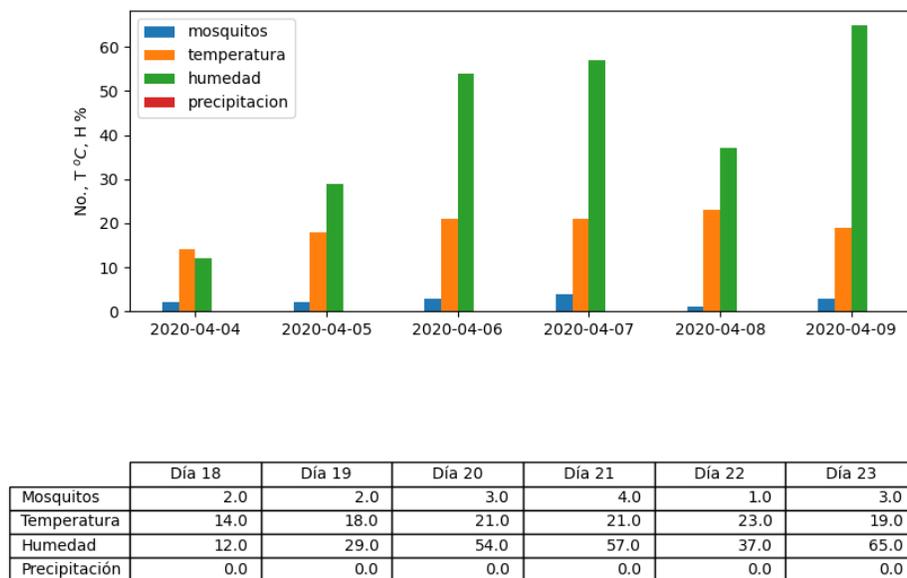


Figura 4.11: Datos correspondientes a la fecha medida de 4 de Abril al 9 de Abril.

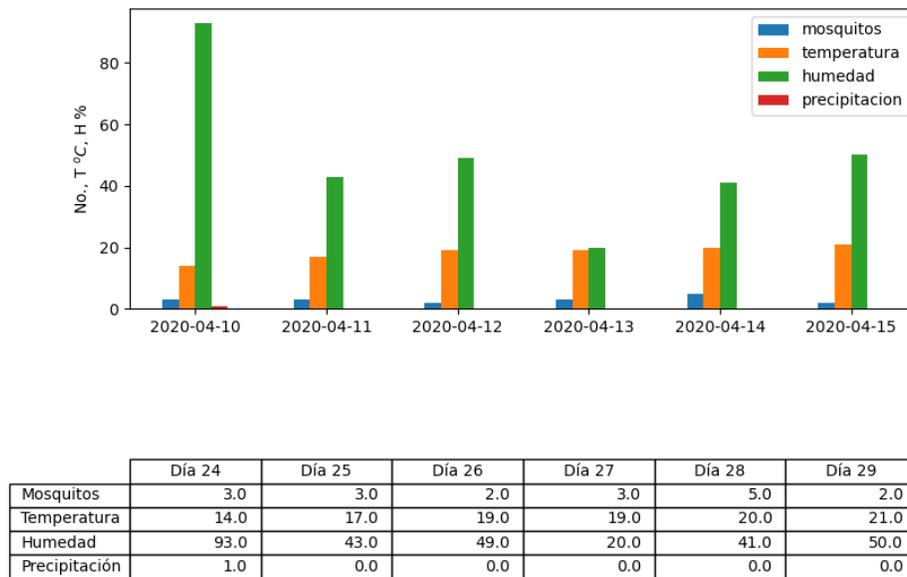


Figura 4.12: Datos correspondientes a la fecha medida de 10 de Abril al 15 de Abril.

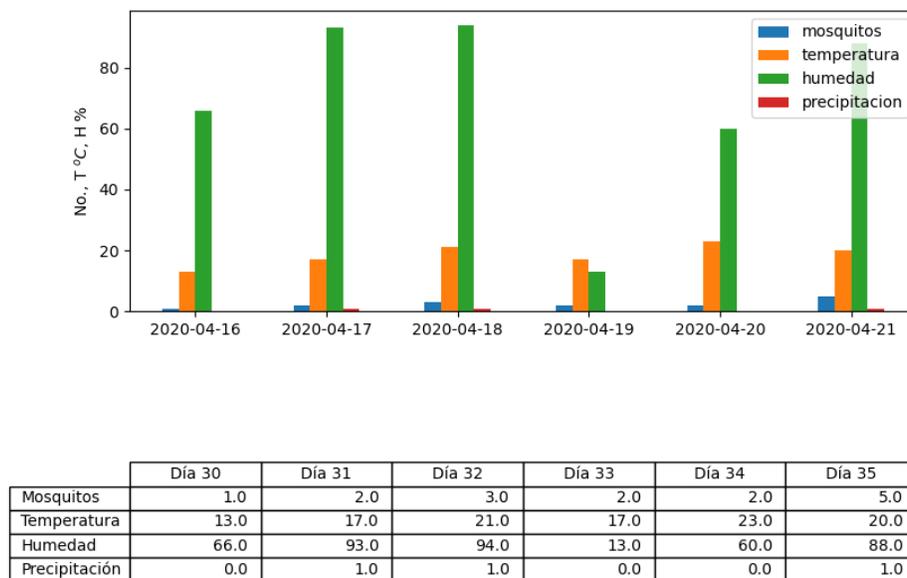


Figura 4.13: Datos correspondientes a la fecha medida de 16 de Abril al 21 de Abril.

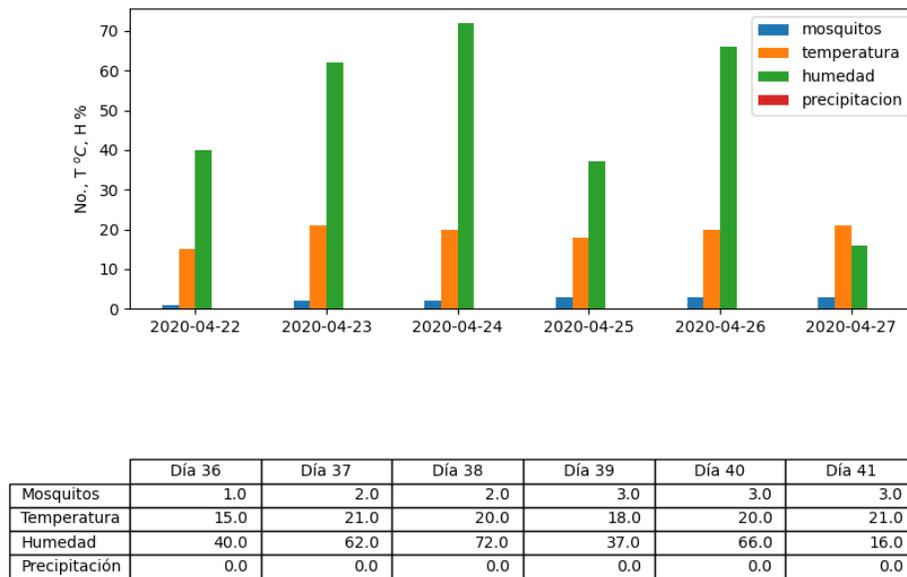


Figura 4.14: Datos correspondientes a la fecha medida de 22 de Abril al 27 de Abril.

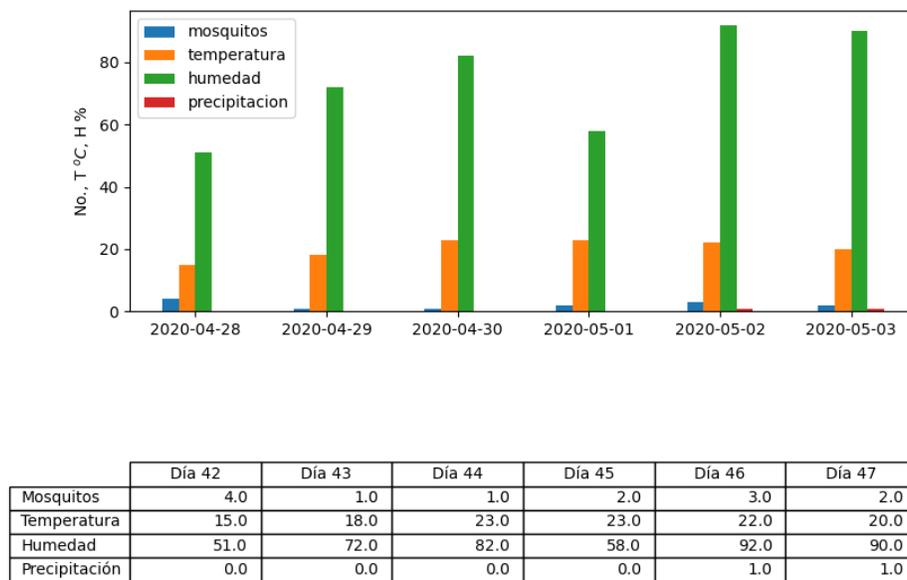


Figura 4.15: Datos correspondientes a la fecha medida de 28 de Abril al 3 de Mayo.

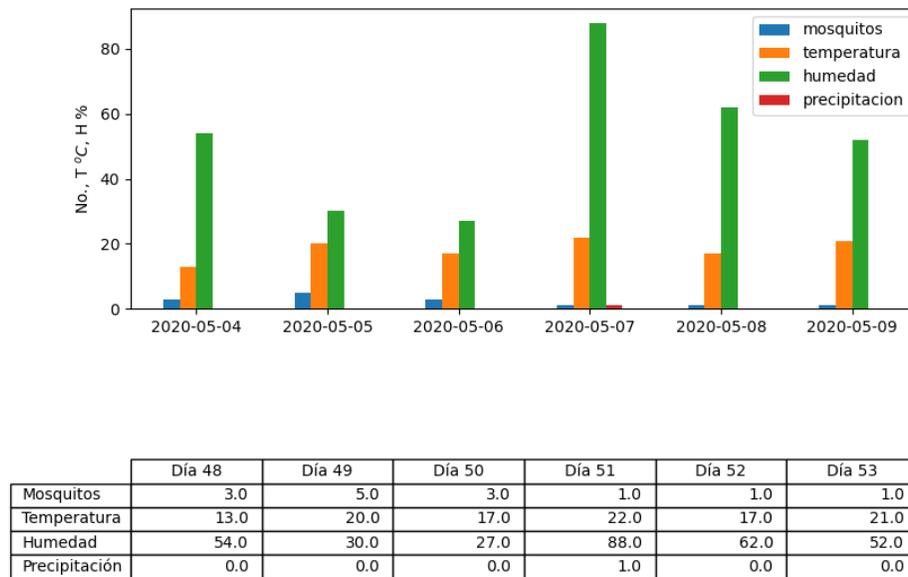


Figura 4.16: Datos correspondientes a la fecha medida de 4 de Mayo al 9 de Mayo.

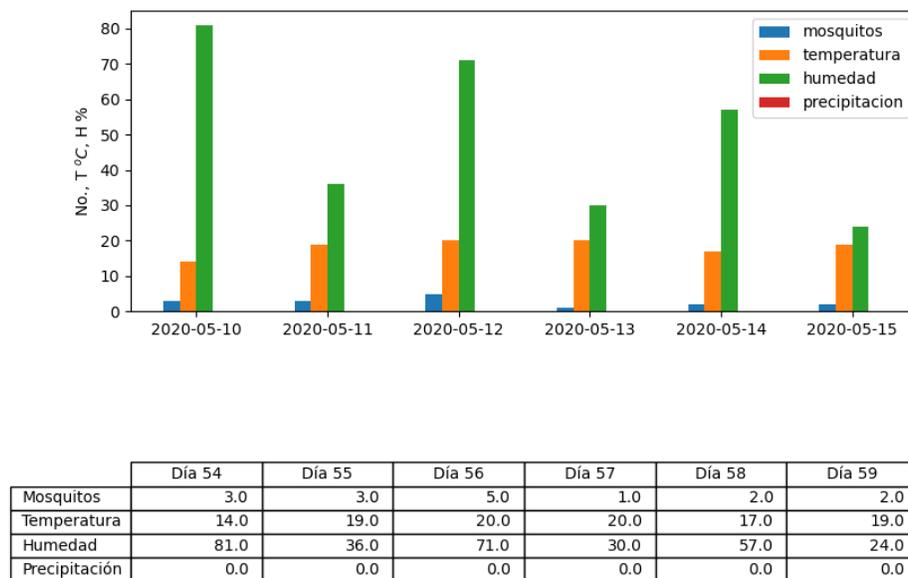


Figura 4.17: Datos correspondientes a la fecha medida de 10 de Mayo al 15 de Mayo.

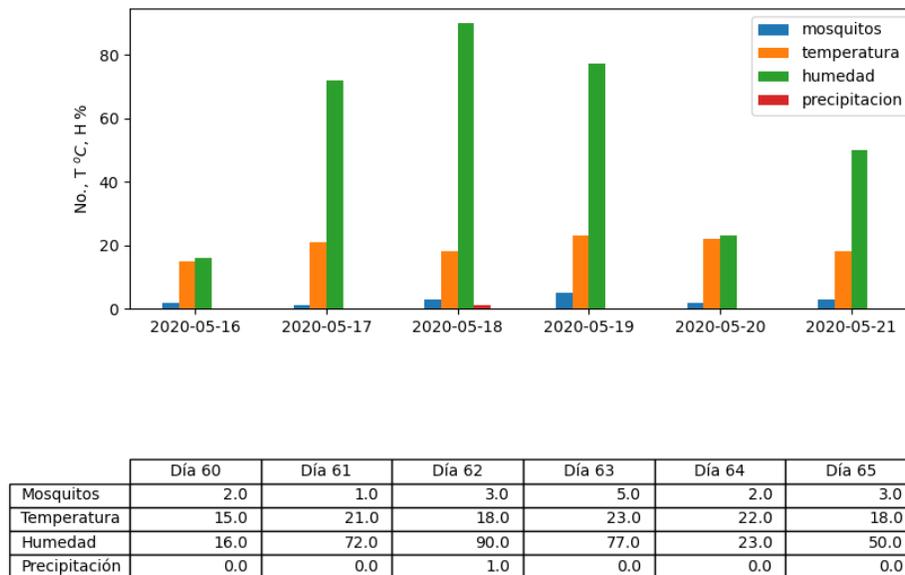


Figura 4.18: Datos correspondientes a la fecha medida de 16 de Mayo al 21 de Mayo.

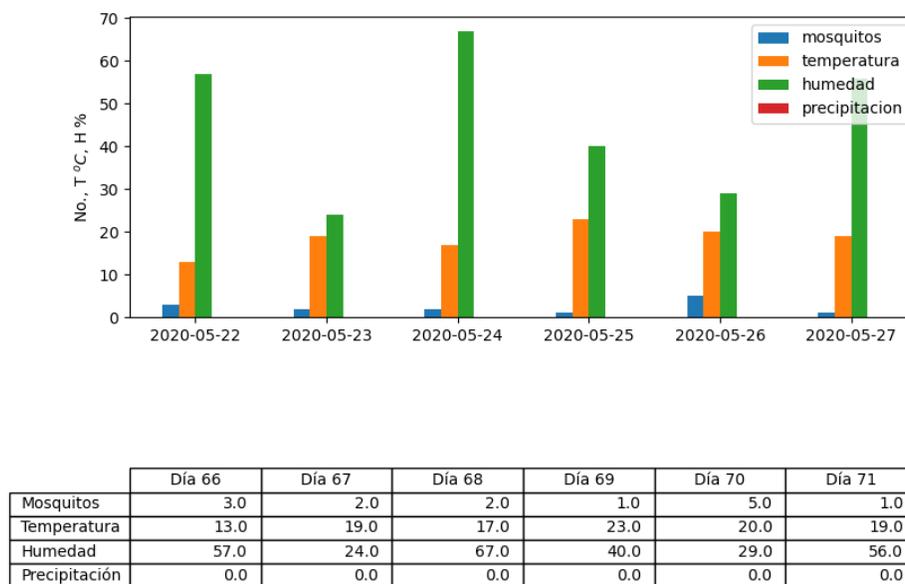


Figura 4.19: Datos correspondientes a la fecha medida de 22 de Mayo al 27 de Mayo.

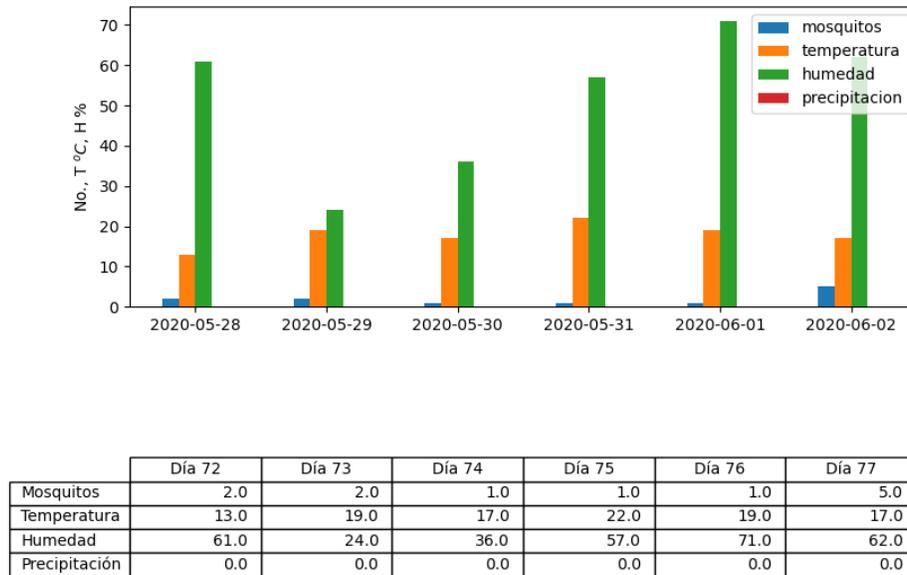


Figura 4.20: Datos correspondientes a la fecha medida de 28 de Mayo al 2 de Junio.

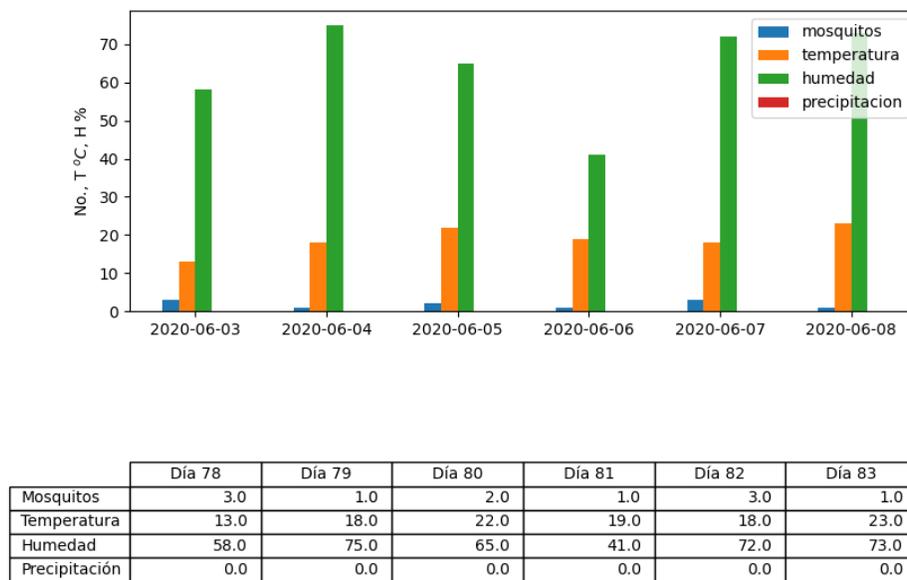


Figura 4.21: Datos correspondientes a la fecha medida de 3 de Junio al 8 de Junio.

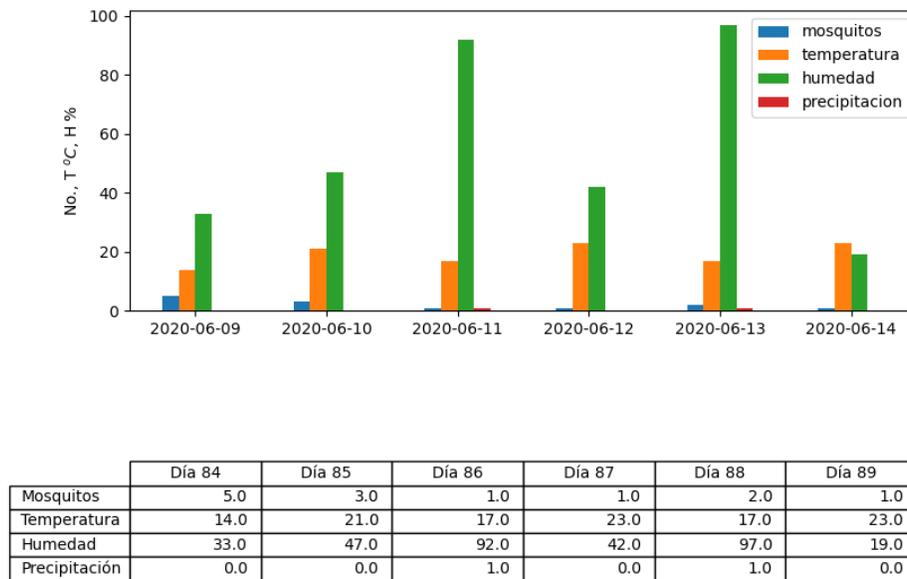


Figura 4.22: Datos correspondientes a la fecha medida de 9 de Junio al 14 de Junio.

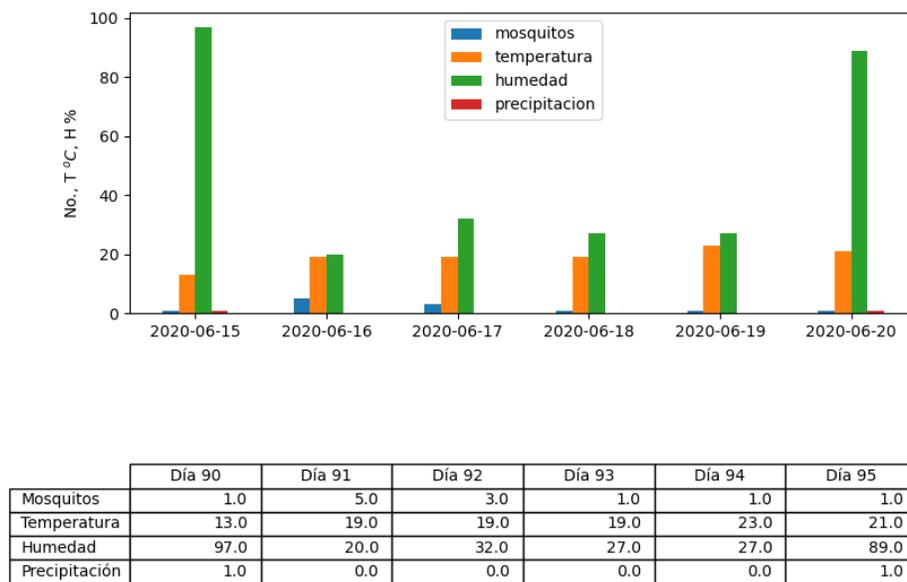


Figura 4.23: Datos correspondientes a la fecha medida de 15 de Junio al 20 de Junio.

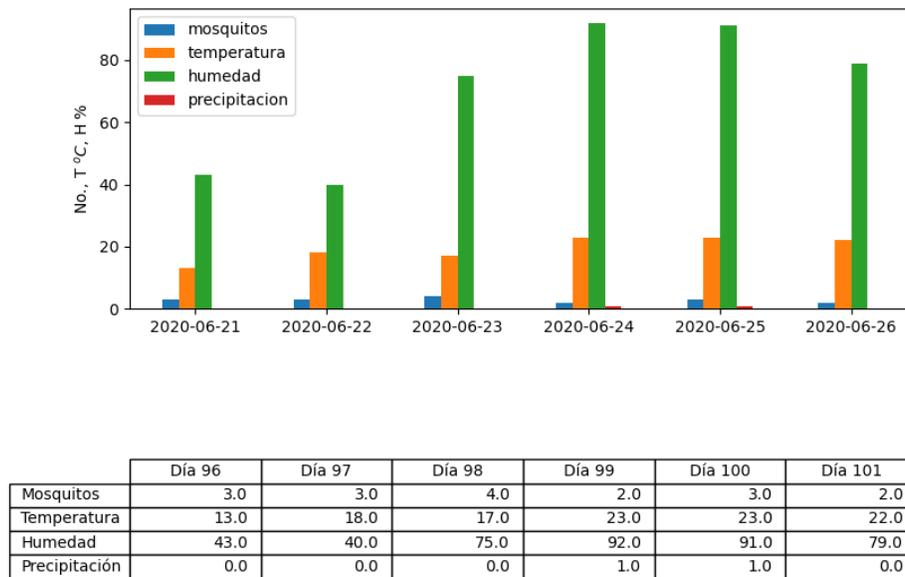


Figura 4.24: Datos correspondientes a la fecha medida de 21 de Junio al 26 de Junio.

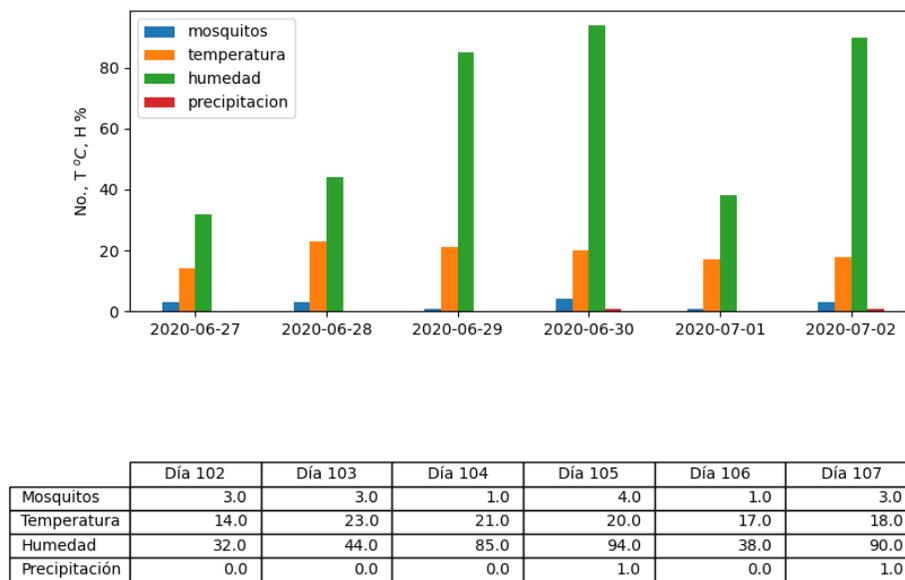


Figura 4.25: Datos correspondientes a la fecha medida de 27 de Junio al 2 de Julio.

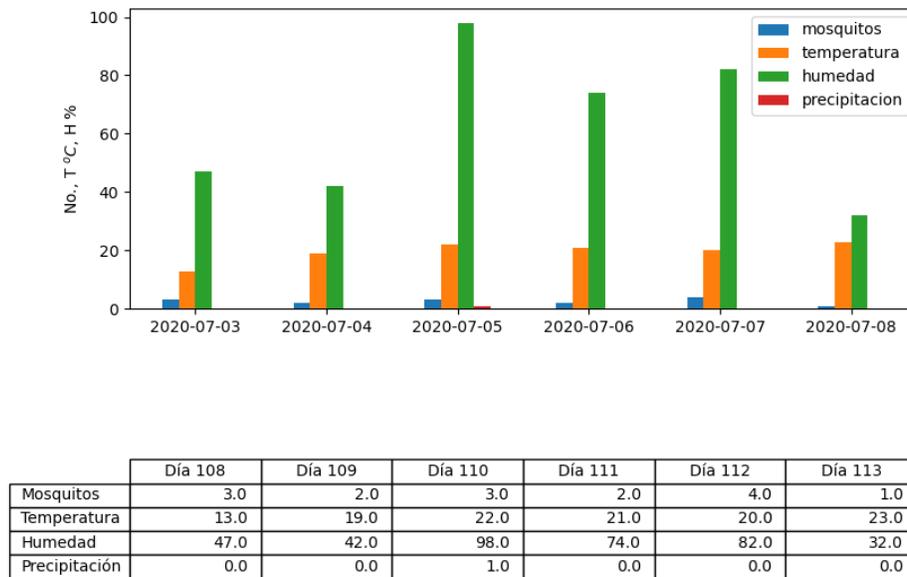


Figura 4.26: Datos correspondientes a la fecha medida de 3 de Julio al 8 de Julio.

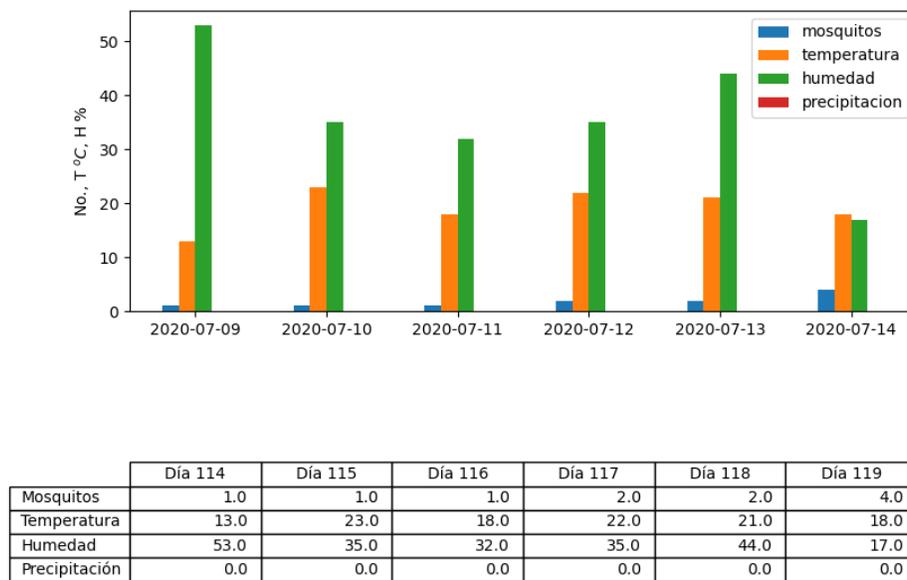


Figura 4.27: Datos correspondientes a la fecha medida de 9 de Julio al 14 de Julio.

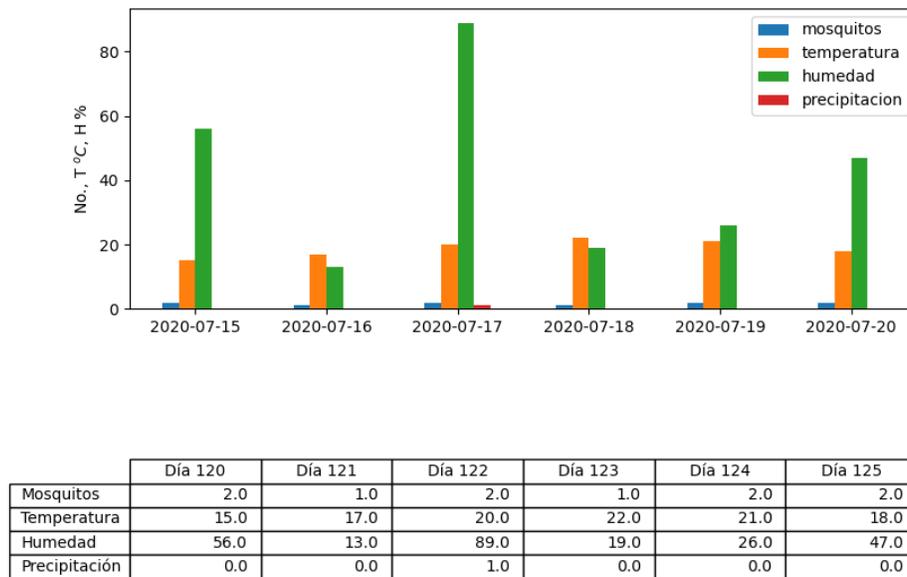


Figura 4.28: Datos correspondientes a la fecha medida de 15 de Julio al 20 de Julio.

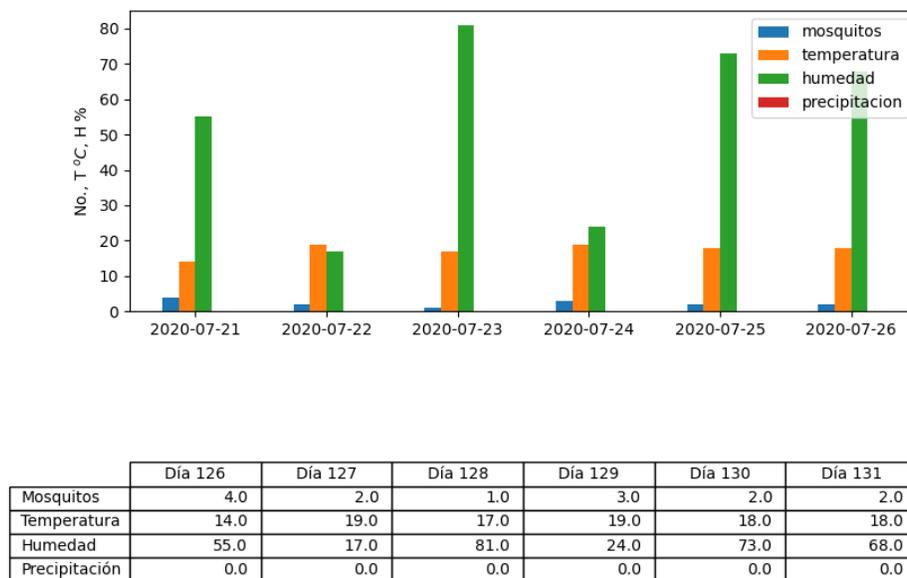


Figura 4.29: Datos correspondientes a la fecha medida de 21 de Julio al 26 de Julio.

4.4. Procesamiento de imagen

En informática, el procesamiento de imágenes digitales se hace uso de una computadora digital para procesar imágenes digitales a través de un algoritmo. Como subcategoría o campo del procesamiento de señales digitales, el procesamiento de imágenes digitales tiene muchas ventajas sobre el procesamiento de imágenes analógicas. Permite que se aplique una gama mucho más amplia de algoritmos a los datos de entrada y puede evitar problemas como la acumulación de ruido y la distorsión durante el procesamiento. Dado que las imágenes se definen en dos dimensiones (quizás más), el procesamiento digital de imágenes puede modelarse en forma de sistemas multidimensionales. La generación y el desarrollo del procesamiento de imágenes digitales se ven afectados principalmente por tres factores: primero, el desarrollo de las computadoras; segundo, el desarrollo de las matemáticas (especialmente la creación y mejora de la teoría discreta de las matemáticas); tercero, ha aumentado la demanda de una amplia gama de aplicaciones en medio ambiente, agricultura, militar, industria y ciencias médicas [41][42].

Uno de los métodos a trabajar y de los cuales se plantearon desde un inicio es el contar mosquitos a partir del procesamiento de imágenes. Este método consiste en convertir, mapear y descomponer una imagen la cual se interpreta en una matriz tridimensional. Python contiene algunas librerías para procesar imágenes, la que se va a utilizar es OpenCv, librería que contiene bastante documentación y contiene algunos libros basados en esta librería.

En la Fig. 4.30 se muestran todas las formas posibles que contiene la librería OpenCv, todas son funciones por partes por ejemplo la función del tipo *binario* se comporta de la siguiente manera

$$dst(x, y) = \begin{cases} maxval & s & src(x, y) > valor, \\ 0 & & cualquier otro, \end{cases} \quad (4.9)$$

de igual manera la función del tipo *hacia cero*

$$dst(x, y) = \begin{cases} src(x, y) & s & src(x, y) > valor, \\ 0 & & cualquier otro, \end{cases} \quad (4.10)$$

para estas funciones el calcular los valores invertidos sólo cambia la función por partes de arriba hacia abajo. Para estas funciones que permiten convertir imágenes a una forma dependiente de la función a elegir, por ejemplo la función del tipo *binario* simplemente pone un cero cuando el *valor* elegido es menor al dato leído (x, y) y pone un uno cuando no se cumple la primera opción. Esto en procesamiento de imágenes al utilizar cualquier función de transformación lo que se obtiene es quitar ruido a la imagen para que esta pueda analizarse de manera más eficiente.

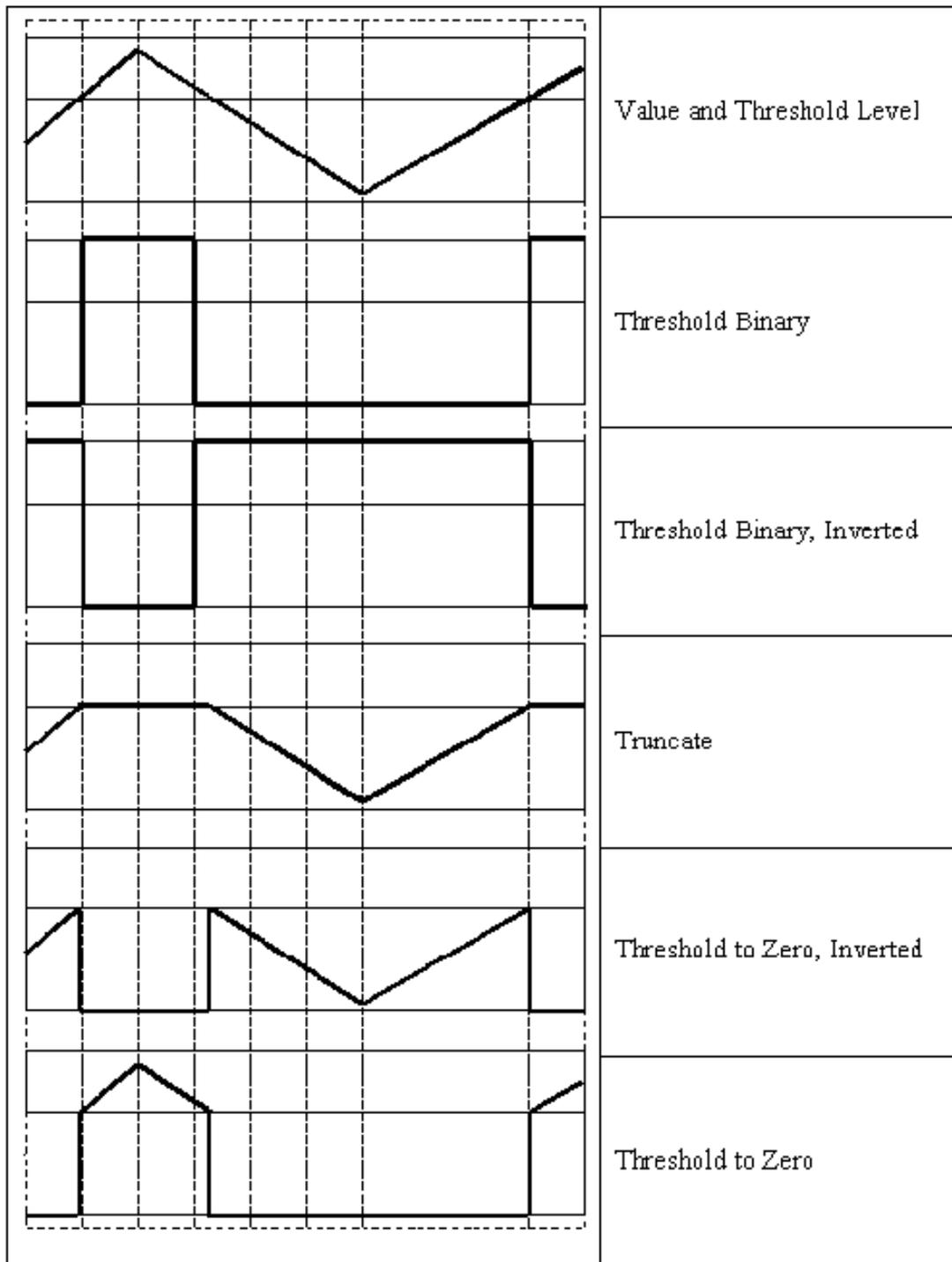
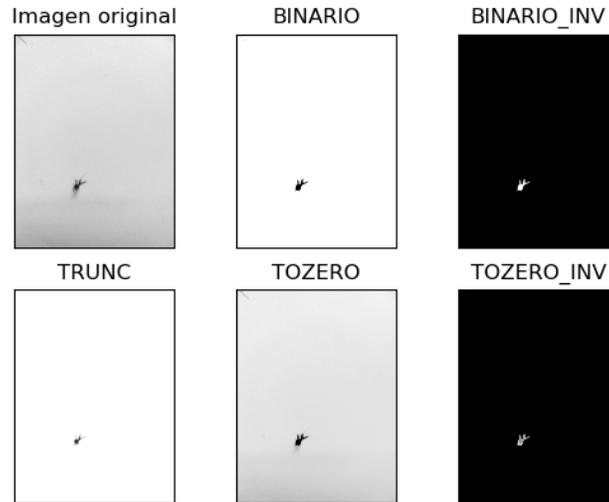
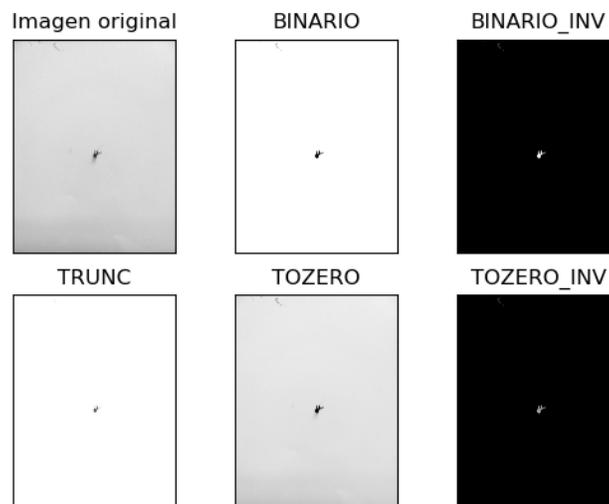


Figura 4.30: Diferentes formas de transformar una imagen.

Las funciones mostradas en la Fig. 4.30 son aplicadas a dos imágenes donde se encuentran a un mosquito en cada imagen para ver la opción que permita identificar un mosquito, como se muestran en las figuras 4.31a y 4.31b.



(a) Mosquito uno.

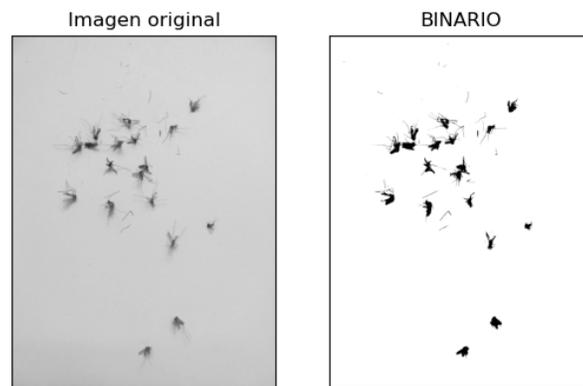


(b) Mosquito dos.

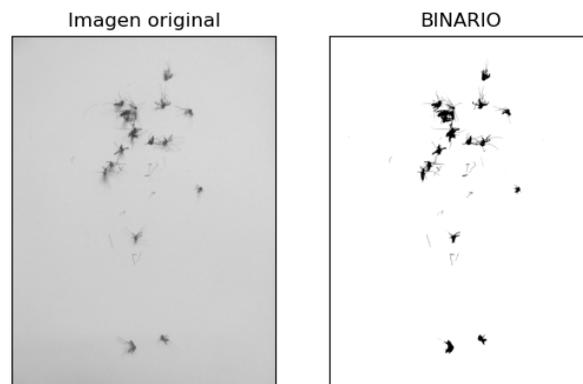
Figura 4.31: Aplicación de funciones de transformación de imagen para un valor = 117.

Anteriormente, se comentó que las transformaciones permiten quitar ruido en las imágenes. Como podemos notar la que mejor se adapta al objetivo es la transformación *binaria* con el valor = 117. Este valor fue encontrado comparando de manera manual un conjunto de imágenes de tal forma que el mosquito no perdiera tantas propiedades. Por ejemplo

las imágenes que estaban alrededor del valor = 50 en el caso de la función binaria la forma del mosquito se perdía mientras que cuando se elegía alrededor de un valor = 134 sucedía algo bastante similar a diferencia de que en valores pequeños la imagen se ponía más oscura mientras que en valores grandes se ponía bastante blanca. El rango de valores permitidos está en el intervalo $[0, 255]$ donde 0 se entiende como negro mientras que 255 corresponde al valor blanco, tal y como se manejan los valores del RGB. Ahora en las figuras 4.32a y 4.32b mostramos a un conjunto de mosquitos convirtiendo la imagen como el ejemplo anterior.



(a) Múltiples mosquitos más separados uno del otro.



(b) Múltiples mosquitos más juntos.

Figura 4.32: Conjunto de mosquitos con distintos grados de separación, en donde se ha aplicado la función de transferencia de imagen binaria para un valor = 117.

El siguiente paso en OpenCv, es aplicar máscaras con objetivo de eliminar cierto ruido. Uno de los más efectivos es el *filtrado Gaussiano*; que es igual que un promediado pero ponderado. Esta ponderación se hace siguiendo la Campana de Gauss. Con esto se consigue dar más importancia a los píxeles que están más cerca del centro de los que están más alejados, ver Fig. 4.33.

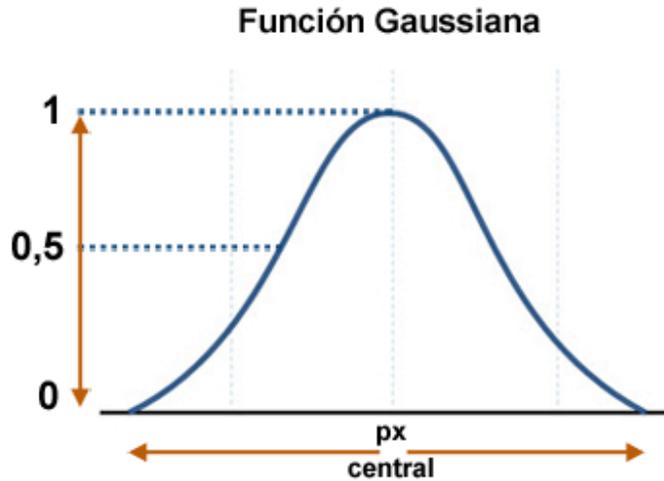


Figura 4.33: Filtro Gaussiano para imágenes.

El objetivo de utilizar una Campana de Gauss conlleva a una aproximación de cómo ve el ojo humano, intentando ser lo más natural posible. La función Gaussiana o Distribución Normal se utiliza en muchas áreas de la ciencia.

Por ejemplo, la altura de la población o el coeficiente intelectual se puede representar a través de la Distribución Normal. Gracias a su forma acampanada y simétrica, hace que los elementos que están más cerca del centro sean los más comunes. Por el contrario, los elementos más lejanos serán los más raros o diferentes.

Para poder aplicarlo en una imagen debemos hacerlo en dos dimensiones. Esto lo hacemos a través de una máscara o kernel de convolución, como se indica en la Fig. 4.34. Como se puede comprobar en la imagen anterior, se da más importancia al píxel central que a los píxeles que están alrededor de éste.

Una vez que se aplicó la función Gaussiana, el siguiente paso es utilizar un detector de bordes. Uno de los grandes inconvenientes de los algoritmos de la visión artificial es la parametrización. Muchas técnicas requieren establecer parámetros o condiciones iniciales según cada situación. En muchas ocasiones esos parámetros son exclusivos para una determinada iluminación o perspectiva.

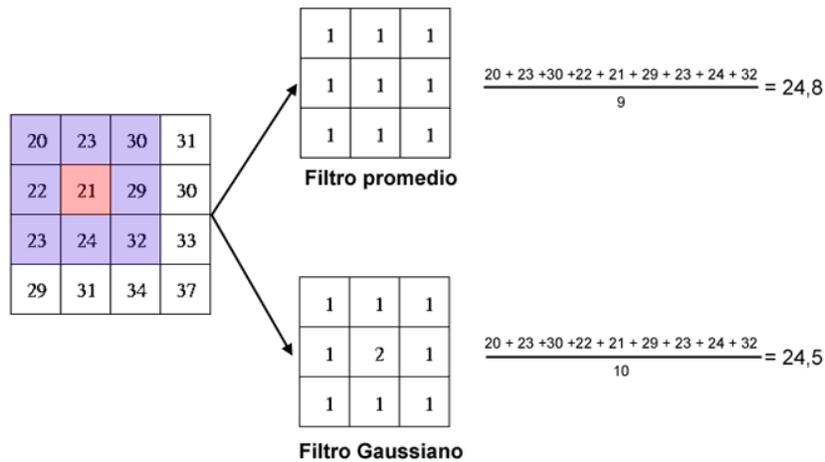


Figura 4.34: Ejemplo Gaussiano para imágenes.

Esto dificulta mucho a la hora de buscar una solución única para diferentes situaciones y por lo tanto, algo que funciona correctamente en unas condiciones de iluminación no tiene porque funcionar en otras circunstancias.

Normalmente, este tipo de algoritmos se entrenan para buscar la mejor solución para múltiples casos. A todo esto se le llama Machine Learning o aprendizaje automático. Este tema es posible abordarlo utilizando lo aprendido en la sección de Redes Neuronales Artificiales pero no se abordará debido a la dificultad que todo esto supone.

Una vez explicado todo lo anterior a esta sección, es momento de utilizar un método proporcionado por OpenCV conocido como Canny el cual permite obtener los bordes de superficies, el cual requiere de un umbral tanto mínimo como máximo, es decir, que tan fino se requiere la detección de bordes. Los resultados de este proceso de obtención de bordes se muestran en las figuras 4.35a a 4.36d. Un dato importante a observar es que cuando los mosquitos suelen estar más separados, el algoritmo creado produce un error de conteo muy pequeño (Figs. 4.35a, 4.35b, 4.35c y 4.35d) pero cuando los mosquitos están más juntos el error de conteo aumenta (4.36a y 4.36b). Por último cuando se tiene muy pocos mosquitos el error de conteo es muy pequeño (Figs. 4.36c y 4.36d). Parte de la programación es aproximar el tamaño de un mosquito por eso es que en la Fig. 4.36d hay una mosca la cual excede el tamaño del mosquito y por tanto ese no es contado. En promedio todos los mosquitos suelen tener un tamaño promediado siempre y cuando la imagen tomada se encuentre a una distancia muy similar. En este caso la altura para tomar fotos fue de 10 cm.



(a) 17 de 17 mosquitos detectados.



(b) 16 de 17 mosquitos detectados.



(c) 18 de 18 mosquitos detectados.



(d) 19 de 17 mosquitos detectados.

Figura 4.35: Mosquitos detectados con procesamiento de imágenes.



(a) 26 de 17 mosquitos detectados.



(b) 29 de 17 mosquitos detectados.



(c) 4 de 3 mosquitos detectados.



(d) 3 de 3 mosquitos detectados.

Figura 4.36: Mosquitos detectados con procesamiento de imágenes.

Capítulo 5

Discusión y Conclusiones

En el capítulo 4 se trabajó en la aplicación y desarrollo del trabajo de tesis. Se trabajaron en múltiples disciplinas lo que permitió completar los objetivos. Las áreas de interés trabajadas fueron: área de electrónica la cual fue apoyada por el profesor M.C. Israel Luna Reyes y la profesora Dr. Mary Carmen Peña donde recibe capacitación y orientación, área de física computacional orientada por el director de tesis.

En las siguientes secciones se discutirán las conclusiones finales del trabajo de investigación realizado en la maestría.

5.1. Dispositivo electrónico

El dispositivo electrónico presenta ciertas partes de construcción, las cuales en seguida se describirán a detalle.

Estructura del dispositivo electrónico. El material que se utilizó fue el MDF para la base y cuerpo del dispositivo electrónico. Desde mi punto de vista y sabiendo que se tiene experiencia en la carpintería, éste es un material muy fácil de trabajar además de que es barato y muy resistente. Este material es bastante bueno para ser trabajado en muchas áreas por ejemplo cuerpos base, máquinas CNC Router, entre otros.

Como todo material, el MDF tiene sus desventajas y para este proyecto la única desventaja en la que nos enfrentamos fue cuando llueve ya que este material suele ser un poco débil, generalmente se hincha y es difícil de que recupere por si solo su estado natural. Para evitar esto es bueno sellar con una combinaciones de materiales derivados de las resinas tales como el poliuretano y catalizador, que al ser mezclados y aplicar al MDF sella el material. También se puede utilizar material de nitrato de celulosa que es más barato y

también sella el material. Es evidente que al utilizar el MDF para el dispositivo electrónico no va durar mucho relativamente por su estructura pero si es bastante bueno para hacer las pruebas ya que una hoja de MDF se tiene mucho material de sobra.

Temperatura y humedad. En el capítulo 2, se describió todos los elementos electrónicos aprendidos. Uno de los objetivos para la construcción del dispositivo electrónico es medir variables climatológicas tales como la temperatura, humedad y precipitación. Para la medición de la temperatura y humedad se utilizó un módulo conocido como DHT22. El módulo DHT22 es un módulo que en un principio ya viene ya calibrado y configurado para devolver dichas mediciones. Sin embargo, para tener una mejor medición es conveniente hacer comparaciones con un sensor analógico para la humedad y otro para la temperatura.

El higrómetro es un instrumento que se utiliza para medir el grado de humedad del aire o de otros gases. El termómetro mide la temperatura. Actualmente, en el mercado existe los dos sistemas de medición en una sola pieza. Esta ventaja es importante ya que podemos resolver la medición de dos variables climatológicas para finalmente los datos obtenidos, resolverlos mediante una aproximación de mínimos cuadrados. Pero debemos tomar en cuenta que estos nuevos datos dependen del sistema analógico a utilizar. Si este sistema por alguna razón éste no mide de manera correcta tendríamos errores, por lo único que nos queda es confiar en el fabricante y que el sistema pasó todas sus pruebas de calidad.

Un dato a considerar del módulo es que en cada envío de medición envía un total de 40 bits en 4 ms. Estos 40 bits corresponden con 2 Bytes para la medición de humedad, 2 Bytes para la medición de temperatura, más un Byte final para la comprobación de errores. La configuración del módulo DHT22 es sencilla para el Arduino, tal y como ya se explicó anteriormente.

La variable de precipitación fue medida manualmente debido a la estructura del dispositivo electrónico. Debido a las condiciones climáticas, en cuestión de lluvia era muy variada durante el día por que al final se decidió contar el dato mientras el dispositivo electrónico esta en funcionamiento.

Método de atracción. Se realizaron distintas pruebas de atracción con los métodos del CO₂ y luz ultravioleta. Cada sistema cuenta con sus ventajas, pero la más factible de utilizar es el método de atracción mediante luz ultravioleta.

Captura de mosquitos. La manera más sencilla de capturarlos es con pegamento de secado lento por ejemplo las tiras atrapa moscas. Los mosquitos y otros insectos simplemente se paran sobre esta tira y quedan atrapados; además se utiliza una ventilador para que estos sean jalados hacia la cinta. Finalmente se cuentan y listo. Para optimizar los gastos de material, se puede utilizar un pegamento atrapamoscas (sticky trap) ya que rinde mucho por la cantidad de uso.

Conteo de mosquitos alternativo. Una forma de contar los mosquitos es llevar esa placa de pegamento al conteo por procesamiento de imágenes, se había pensado en colocar una cámara e instalarla en el dispositivo electrónico, pero se concluyó que no es tan factible ya que es muy costoso y como se explicó en la sección 4.4, es difícil contar mosquitos con una calidad de cámara de resolución baja. Por lo que entre más calidad de cámara es más costosa.

Una cámara que esté destinada para ser integrada en el dispositivo electrónico, ya que se encuentra expuesta y posiblemente pueda ser vandalizada lo que perderíamos dinero por el simple hecho de poner una cámara de calidad. Por esto, es preferible tener una cámara apartada, fija y en un lugar seguro donde sólo sea utilizada para tomar fotografías para contar los mosquitos.

5.2. Ecuación de reacción-difusión

Para resolver la ecuación de reacción-difusión se hace mediante diferencias finitas tal y como se describió en la sección 4.1. Esta ecuación se adapta bastante bien para simular de manera numérica la propagación de mosquitos respecto a la posición de las fuentes de reproducción de mosquitos.

La ecuación de reacción-difusión sólo se resolvió de manera numérica para coeficientes de reacción y difusión constantes. Sin embargo, esto es suficiente para comprender la forma en la que se simula la propagación de mosquitos donde el centro de la función donde se tiene un máximo que representa la mayor concentración de mosquitos y conforme la ecuación se difunde se tiene menos concentración de mosquitos. Entonces si colocamos distintos dispositivos electrónicos dentro del dominio podemos obtener los valores de la concentración dentro de la ubicación de cada uno de ellos. Evidentemente los dispositivos electrónicos que estén más cerca al punto de concentración de la simulación numérica se obtendrá más concentración que aquellas que están más alejadas.

Generar múltiples simulaciones numéricas permite hacer uso de ellas tal y como se resolvió en la tesis de licenciatura en *Predicción de la posición de fuentes de reproducción de mosquitos mediante el uso de dispositivos electrónicos e inteligencia artificial*. Básicamente ya se tiene un modelo de predicción de mosquitos en la Fig. 5.1 se muestra la metodología que se siguió para clasificar la información.

Se tenían 81 simulaciones numéricas que están representadas por círculos negros y blancos, aquí cada simulación representa el mayor punto de concentración de la propagación de mosquitos. El dominio se dividió en 25 subregiones y en cada subregión cuenta con un cierto número de simulaciones numéricas. Las que subregiones que tienen menos simulaciones se puede entender que es un lugar seco donde la reproducción de mosquitos es menor que en las subregiones que cuentan más de ellas. Entonces hasta aquí lo único

Dominio de épocas

○ 1 ○ ○ ●	○ 6 ○ ● ○	○ 11 ○ ○ ●	○ 16 ○ ● ○	○ 21 ○ ● ○
○ 2 ● ○ ○	● 7 ○ ○ ○	○ 12 ● ○ ○	● 17 ○ ○ ○	● 22 ○ ○ ○
○ 3 ● ○ ○	○ 8 ○ ● ○	○ 13 ○ ○ ●	● 18 ○ ○ ○	● 23 ○ ● ○
○ 4 ○ ○ ●	○ 9 ○ ● ○	○ 14 ○ ○ ●	○ 19 ○ ● ○	○ 24 ○ ● ○
○ 5 ● ○ ○	● 10 ○ ○ ○	○ 15 ● ○ ○	● 20 ○ ○ ○	○ 25 ○ ○ ○

Figura 5.1: Ejemplo de clasificación de datos que se trabajó.

que se hizo fue ordenar la información para que al momento de llevar a la clasificación de las redes neuronales pueda ser más fácil para aproximar la posición de las fuentes de reproducción de mosquitos. Los círculos blancos son aquellos que se clasificarán mientras que los círculos negros se utilizará para probar la efectividad de clasificación de la red neuronal. En la sección 3.3.4 se muestra de manera general el funcionamiento de una red neuronal y como es que se entrena para clasificar. Esta es la forma en la que creó el modelo de predicción de la posición de fuentes de reproducción de mosquitos.

5.3. Perspectiva

Uno de los aspectos que se puede resolver es encontrar la correlación entre nuestros modelos teóricos tanto simulación numérica, el modelo de predicción de fuentes de reproducción de mosquitos y datos reales medidos por el dispositivo electrónico, aunque hasta ahora no fue posible debido a que no se cuenta con los suficientes dispositivos electrónicos.

Como podemos notar ya se cuenta con modelos teóricos y un dispositivo electrónico que cuenta mosquitos. Lo que se puede mejorar es que se tienen por ambas partes, por un lado, en lo que concierne a la ecuación de reacción-difusión es resolver simulaciones numéricas variando los parámetros para los coeficientes de reacción y difusión en relación de las variables climatológicas como la temperatura y humedad, y por el lado, el dispositivo electrónico se puede mejorar para hacerlo más eficiente en el sentido de contar mosquitos, por otro lado permitirá tener un mayor control de enfermedades y que pueda medir más variables climatológicas como la precipitación, la dirección y velocidad del viento de manera automática. Para el conteo de mosquitos se puede mejorar la identificación y con-

teo de mosquitos por el procesamiento de imagen utilizando herramientas de inteligencia artificial como las redes neuronales artificiales.

Apéndice A

Sistema de numeración

El sistema de numeración donde la mayoría de las personas nos comunicamos es en el sistema decimal. ¿Por qué desempeña un papel tan privilegiado precisamente el número 10? Una persona ajena a estas cuestiones contestaría quizá sin vacilar que se trata sencillamente de que el número 10 es redondo y resulta cómodo multiplicar por él cualquier número, así como contar por decenas, centenares, etc. Sin embargo, la situación es precisamente la opuesta: el número 10 es redondo debido a que se toma por base del sistema de numeración. Si pasamos a otro sistema de numeración, digamos septenario (en el que este número se representa como $(13)_7$), dejará inmediatamente de ser redondeado. Las razones por las cuales precisamente el sistema decimal ha sido universalmente aceptado no son, ni mucho menos, de índole matemática: los diez dedos de las manos han constituido el aparato primario de cálculo que empleó el hombre desde los tiempos prehistóricos. Valiéndose de los dedos es fácil contar hasta diez. Al llegar a diez, es decir, después de consumir todas las posibilidades de nuestro “aparato de cálculo” natural, lo lógico es considerar el número 10 como una unidad nueva, mayor (la unidad del orden siguiente). Diez decenas forman la unidad del tercer orden y así sucesivamente. Por lo tanto, precisamente el cálculo a base de los dedos de las manos ha dado origen al sistema que nos parece ahora completamente natural [51].

El número 2 es el menor de los números que se puede tomar como base de un sistema de numeración. El sistema correspondiente a esta base, llamado binario, es muy antiguo. Lo empleaban, aunque de forma muy imperfecta, algunas tribus de Australia y Polinesia. La ventaja de este sistema es su extrema sencillez. En el sistema binario intervienen sólo dos cifras 0 y 1; el número 2 representa la unidad del orden siguiente. También son muy sencillas las reglas de las operaciones con los números escritos en el sistema binario. Las reglas principales de adición y multiplicación se muestran se resumen en la tabla A.1 [52].

Por otro lado, en matemáticas y computación, el hexadecimal (también base 16 o hexadecimal) es un sistema posicional que representa números usando una base de 16. A diferencia de la forma común de representar números con diez símbolos, usa dieciséis

Adición	Multiplicación		
$0 + 0 = 0$		0	1
$0 + 1 = 1$	0	0	0
$1 + 1 = (10)_2$	1	0	1

Tabla A.1: Adición y multiplicación en el sistema binario.

símbolos distintos, con mayor frecuencia los símbolos “0 - 9” para representar valores de cero a nueve, y “A - F” (o alternativamente “a - f”) para representar valores de diez a quince. Los numeradores hexadecimales son ampliamente utilizados por los diseñadores y programadores de sistemas informáticos, ya que proporcionan una representación amigable para los humanos de los valores codificados en binario. Cada dígito hexadecimal representa cuatro dígitos binarios, también conocidos como mordiscos, que es medio byte. Por ejemplo, un sólo byte puede tener valores que van desde 00000000 a 11111111 en forma binaria, que se pueden representar convenientemente como 00 a FF en hexadecimal [53].

A.1. Conversiones

Se han inventado métodos para convertir algún número de algún sistema de numeración, dicho de otro modo, al tener un número el binario es posible convertirlo a decimal o su defecto hexadecimal, también al tener un número en hexadecimal es posible convertirlo a otros sistemas de numeración.

Para convertir números decimales a binario o hexadecimal y viceversa se aplican las siguientes reglas:

1. Número Decimal a Binario. Para hacer la conversión de decimal a binario, hay que ir dividiendo el número decimal entre dos y anotar en una columna a la derecha el resto (un 0 si el resultado de la división es par y un 1 si es impar).

La lista de ceros y unos leídos de abajo a arriba es el resultado.

Por ejemplo: se quiere convertir a binario el número $(79)_{10}$

- a) 79 Dividimos entre dos: 1 (impar).
- b) 39 Dividimos entre dos: 1 (impar).
- c) 19 Dividimos entre dos: 1 (impar).
- d) 9 Dividimos entre dos: 1 (impar).
- e) 4 Dividimos entre dos: 0 (par).
- f) 2 Dividimos entre dos: 0 (par).

- g) 1 1 (impar).
 Por tanto, $(79)_{10} = (1001111)_2$.
 Otro ejemplo visual.

28	2				
0	14	2			
	0	7	2		
		1	3	2	
			1	1	

Por tanto, $(28)_{10} = (11100)_2$

2. Número Binario a Decimal. Una forma de hacer la conversión de binario a decimal, se tiene que dividir el número binario en celdas de 4 y generalizarlo como se muestra en la tabla A.2.

Forma Binaria representación												
B_m	...	B_9		B_8	B_7	B_6	B_5		B_4	B_3	B_2	B_1
2^n	...	2^8		2^7	2^6	2^5	2^4		2^3	2^2	2^1	2^0

Tabla A.2: Composición de números binarios.

Tomemos el ejemplo anterior donde se sabe que $(1001111)_2 = (79)_{10}$, como se observa este número binario esta compuesto de 7 números (ceros y unos). Entonces, colocando este número en la tabla A.2, enseguida se resuelve el valor de la casilla de abajo sólo en donde aparecen en 1 y finalmente se suman los valores, por lo que se deberá obtener el valor decimal. Sea:

	1	0	0		1	1	1	1
2^7	2^6	2^5	2^4		2^3	2^2	2^1	2^0

Por tanto $(79)_{10} = 64 + 0 + 0 + 8 + 4 + 2 + 1$.

3. Número Decimal a Hexadecimal. utiliza el número 16 como su base (radix). Como sistema de numeración de base 16, utiliza 16 símbolos. Estos son los 10 dígitos decimales (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) y las primeras seis letras del alfabeto inglés (A, B, C, D, E, F). Las letras se usan debido a la necesidad de representar los valores 10, 11, 12, 13, 14 y 15 cada uno en un sólo símbolo. El hexadecimal se usa en matemáticas y tecnologías de la información como una forma más amigable de representar números binarios. Cada dígito hexadecimal representa cuatro dígitos binarios; por lo tanto, hexadecimal es un lenguaje para escribir binario en forma abreviada.

La conversión de decimal a hexadecimal se puede lograr aplicando la división repetida y el algoritmo restante. En pocas palabras, el número decimal se divide repetidamente por la raíz 16. Entre estas divisiones, el resto da el equivalente hexadecimal en orden inverso.

Aquí es cómo convertir decimal a hexadecimal paso a paso:

- Paso 1: Si el número decimal dado es menor que 16, el equivalente hexadecimal es el mismo. Recordando que las letras A, B, C, D, E y F se usan para los valores 10, 11, 12, 13, 14 y 15, se convierten en consecuencia. Por ejemplo, el número decimal 15 será F en hexadecimal.
- Paso 2: Si el número decimal dado es 16 o mayor, divida el número por 16.
- Paso 3: Escribe el resto.
- Paso 4: Divide la parte anterior al punto decimal de tu cociente entre 16 nuevamente. Escribe el resto.
- Paso 5: Continúa este proceso de dividir entre 16 y anotar los restos hasta que el último dígito decimal que te quede sea menor que 16.
- Paso 6: Cuando el último dígito decimal es menor que 16, el cociente será menor que 0 y el resto será el dígito mismo.
- Paso 7: El último resto que obtenga será el dígito más significativo de su valor hexadecimal, mientras que el primer resto del Paso 3 es el dígito menos significativo. Por lo tanto, cuando escribe los restos en orden inverso, comenzando en la parte inferior con el dígito más significativo y yendo hacia la parte superior, alcanzará el valor hexadecimal del número decimal dado.

Ahora, apliquemos estos pasos, por ejemplo, al número decimal $(501)_{10}$

- Paso 1: Como 501 es mayor que 16, divide entre 16. $501/16 = 31.3125$
- Paso 2: Para calcular el resto, debe multiplicar la parte después del punto decimal por 16. $0.3125 * 16 = 5$
Entonces, el primer resto (y el dígito menos significativo en hexadecimal) es 5.
- Paso 3: Divida 31 (la parte del cociente que está antes del punto decimal) por 16. $31/16 = 1.9375$
- Paso 4: Calcule el resto. $0.9375 * 16 = 15$
- Paso 5: Divida la parte entera del último cociente entre 16. $1/16 = 0.0625$
- Paso 6: Calcule el resto. $0.0625 * 16 = 1$
- Paso 7: Los restos escritos de abajo hacia arriba le dan los valores hexadecimales 1, 15 y 5. Como 15 es igual a F en los números hexadecimales, $(501)_{10} = (1F5)_{16}$

4. Número Hexadecimal (Hex) a Decimal. Hex es un número base 16 y el decimal es un número base 10. Necesitamos saber el equivalente decimal de cada dígito de número hexadecimal. Antes que se debe de consultar la tabla A.3 para poder verificar el gráfico de hexadecimal a decimal.

Estos son los pasos para convertir hexadecimal a decimal:

- Obtenga el equivalente decimal del hexadecimal de la tabla A.3.
- Multiplique cada dígito con 16 potencias de ubicación de dígitos. (basado en cero, 7DE: la ubicación E es 0, la ubicación D es 1 y la ubicación 7 es 2)

Hexadecimal	Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Tabla A.3: Configuración de símbolos Hex a decimales.

- Suma todos los multiplicadores.

Se muestra un ejemplo:

- $7DE$ es un número hexadecimal
- $7DE = (7 * 16^2) + (13 * 16^1) + (14 * 16^0)$
- $7DE = (7 * 256) + (13 * 16) + (14 * 1)$
- $7DE = 1792 + 208 + 14$
- $7DE = 2014$ (en número decimal)

5. Número Binario a Hexadecimal. Para empezar, se necesita hacer agrupaciones de 4 dígitos en binario. Si por lo que sea no se tiene 4 celdas para hacer un grupo de cuatro, añade tantos ceros a la izquierda como necesites. Por ejemplo, imagina que queremos convertir a hexadecimal el siguiente código binario:

1101001100101.

Haciendo las agrupaciones de 4 dígitos (de derecha a izquierda) nos queda la cadena organizada de la siguiente forma:

0001 1010 0110 0101.

Notamos que, nos ha tocado añadir 3 ceros al último grupo (el de la izquierda del todo). Con las agrupaciones ya hechas, aplicaremos un pequeño “truco” para aprender a convertir de binario a hexadecimal. Este truco consiste en añadir una especie

de exponente encima de cada dígito del grupo de cuatro. Siempre llevan la misma colocación y es la siguiente:

$$0^8 0^4 0^2 1^1 / 1^8 0^4 1^2 0^1 / 0^8 1^4 1^2 0^1 / 0^8 1^4 0^2 1^1.$$

De izquierda a derecha, siempre colocaremos:

- Un 8 en el primer dígito.
- Un 4 en el segundo.
- Un 2 en el tercero.
- Un 1 en el cuarto.

Estos números que hemos colocado representan la cantidad de 8's, 4's, 2's y 1's que tenemos en cada grupo. Por ejemplo, si cogemos el conjunto:

$$0^8 1^4 0^2 1^1.$$

Vemos que al sumar cada dígito

$$0^8 + 1^4 + 0^2 + 1^1 = 5.$$

Por lo tanto, el grupo 0101 en binario es igual a 5 en hexadecimal. Ahora tenemos que repetir lo mismo con cada grupo:

$$0^8 0^4 0^2 1^1 = 0 + 0 + 0 + 1 = 1$$

$$1^8 0^4 1^2 0^1 = 8 + 0 + 2 + 0 = 10$$

$$0^8 1^4 1^2 0^1 = 0 + 4 + 2 + 0 = 6$$

$$0^8 1^4 0^2 1^1 = 0 + 4 + 0 + 1 = 5$$

Ya hemos pasado el número binario a hexadecimal y utilizando la tabla A.3. Determinamos que ese 10 que hemos obtenido en el paso anterior es en realidad una letra A.

Por tanto $(1101001100101)_2 = (1A65)_{16}$.

6. Número Hexadecimal a Binario. En el sistema hexadecimal podemos escribir números como 3ADF3, 1A1B1C, B31, DAB, etc, es decir, es un sistema de números y letras (con base de dieciséis) cuyos posibles valores numéricos y letras pueden ser (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) para números y (A,B,C,D,E,F) para letras por cada valor posicional, mientras que en el caso del sistema binario sólo podemos escribir números como 01100111, 1110, 011, 1, etc, es decir, números (con base de dos) cuyos únicos dos posibles valores (0 y 1) por cada valor posicional / dígito.

El sistema hexadecimal a binario es el proceso por el cual convertiremos cualquier número hexadecimal en un número binario. La tabla A.4 tenemos la forma en la que se maneja cada símbolo y valor numérico en Hexadecimal y Binario.

Hexadecimal	0	1	2	3	4	5	6	7
Binario	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal	8	9	A	B	C	D	E	F
Binario	1000	1001	1010	1011	1100	1101	1110	1111

Tabla A.4: Sistema de conversión de hexadecimal a binario y viceversa.

En este proceso habremos creado una cadena de valores numéricos de CEROS (0) y UNOS (1) que al agruparlos (y eliminando los ceros de la izquierda en caso de que existan) obtendremos el número binario correspondiente al número hexadecimal que hayamos querido convertir.

Como puedes comprobar el proceso para realizar la conversión numérica hexadecimal a binaria es muy sencilla, por ejemplo

Hexadecimal	A4F0C				
	A	4	F	0	C
Binario	1010	0100	1111	0000	1100
	$(A4F0C)_{16} = (10100100111100001100)_2$				

A.2. Código ASCII

ASCII significa Código Estándar Estadounidense para el intercambio de información. Las computadoras sólo pueden entender números, por lo que un código ASCII es la representación numérica de un carácter como 'a' ó '@' ó una acción de algún tipo. ASCII se desarrolló hace mucho tiempo y ahora los caracteres que no se imprimen rara vez se usan para su propósito original. A continuación se muestra la tabla de caracteres ASCII y esto incluye descripciones de los primeros 32 caracteres que no se imprimen. Para almacenar números y guardar el texto en su memoria, los ordenadores deben convertir los grafemas en números. En la primera ASCII había una correspondencia entre los números y grafema básica de la ortografía inglesa, esto funcionaba hasta que el número llegó a 127, que corresponde a un código de 7 bits, ya que las computadoras funcionan en sistema de conteo binario, y se convierte en notación binaria, 127 equivalente a un número de 7 dígitos. Posteriormente, el número ASCII se amplió a 255. En la notación binaria, esto corresponde a un número de 8 dígitos, y por lo tanto se considera un código de 8 bits [55].

Actualmente existen muchos sistemas numéricos que se utilizan en ciertas ramas de las ciencias debido a que ese sistema numérico es mucho más eficiente de usar. Para los objetivos del proyecto basta con saber el sistema numérico decimal, hexadecimal y binario, además de conocer los métodos para poder convertir de un sistema a otro.

TABLA DE CARACTERES DEL CÓDIGO ASCII

1	⊙	25	↓	49	1	73	I	97	a	121	y	145	⌘	169	~	193	⌞	217	⌞	241	+
2	●	26		50	2	74	J	98	b	122	z	146	⌘	170	⌞	194	⌞	218	⌞	242	IV
3	♥	27		51	3	75	K	99	c	123	{	147	ò	171	⌞	195	⌞	219	⌞	243	V
4	♦	28	←	52	4	76	L	100	d	124		148	ò	172	⌞	196	⌞	220	⌞	244	⌞
5	▲	29	↔	53	5	77	M	101	e	125	}	149	ò	173	⌞	197	⌞	221	⌞	245	⌞
6	◆	30	▲	54	6	78	N	102	f	126	~	150	ù	174	⌞	198	⌞	222	⌞	246	⌞
7		31	▼	55	7	79	O	103	g	127	~	151	ù	175	⌞	199	⌞	223	⌞	247	⌞
8		32		56	8	80	P	104	h	128	~	152	ÿ	176	⌞	200	⌞	224	⌞	248	⌞
9		33	!	57	9	81	Q	105	i	129	~	153	Û	177	⌞	201	⌞	225	⌞	249	⌞
10		34	"	58	:	82	R	106	j	130	~	154	Û	178	⌞	202	⌞	226	⌞	250	⌞
11		35	#	59	;	83	S	107	k	131	~	155	ç	179	⌞	203	⌞	227	⌞	251	⌞
12		36	\$	60	<	84	T	108	l	132	~	156	£	180	⌞	204	⌞	228	⌞	252	⌞
13		37	%	61	=	85	U	109	m	133	~	157	¥	181	⌞	205	⌞	229	⌞	253	⌞
14		38	&	62	>	86	V	110	n	134	~	158	¥	182	⌞	206	⌞	230	⌞	254	⌞
15		39	'	63	?	87	W	111	o	135	~	159	ƒ	183	⌞	207	⌞	231	⌞	255	⌞
16	*	40	(64	@	88	X	112	p	136	~	160	á	184	⌞	208	⌞	232	⌞	255	⌞
17		41)	65	A	89	Y	113	q	137	~	161	í	185	⌞	209	⌞	233	⌞	255	⌞
18	‡	42	*	66	B	90	Z	114	r	138	~	162	ó	186	⌞	210	⌞	234	⌞	255	⌞
19	‡‡	43	+	67	C	91	[115	s	139	~	163	ú	187	⌞	211	⌞	235	⌞	255	⌞
20	‡‡‡	44	,	68	D	92	\	116	t	140	~	164	ñ	188	⌞	212	⌞	236	⌞	255	⌞
21	‡‡‡‡	45	-	69	E	93]	117	u	141	~	165	Ñ	189	⌞	213	⌞	237	⌞	255	⌞
22	‡‡‡‡‡	46	.	70	F	94	^	118	v	142	~	166	•	190	⌞	214	⌞	238	⌞	255	⌞
23	‡‡‡‡‡‡	47	/	71	G	95	_	119	w	143	~	167	°	191	⌞	215	⌞	239	⌞	255	⌞
24	‡‡‡‡‡‡‡	48	0	72	H	96	̀	120	x	144	~	168	¿	192	⌞	216	⌞	240	⌞	255	⌞

www.rej-dec.com

PRESIONA LA TECLA
Alt
 MÁS EL NUMERO
 CORTESÍA DE


Bibliografía

- [1] <https://www.isglobal.org/-/mosquito-el-animal-mas-letal-del-mundo>
- [2] “Malaria Fact sheet N° 94”. WHO. March 2014. Archived from the original on 3 September 2014. Retrieved 28 August 2014.
- [3] Caraballo H (2014). “Emergency department management of mosquito-borne illness: Malaria, dengue, and west nile virus”. *Emergency Medicine Practice*. 16 (5).
- [4] Tan SY, Sung H (2008). “Carlos Juan Finlay (1833–1915): Of mosquitoes and yellow fever” (PDF). *Singapore Medical Journal*. 49 (5): 370–71. PMID 18465043.
- [5] Solomon T, Dung NM, Kneen R, Gainsborough M, Vaughn DW, Khanh VT (2000). “Japanese encephalitis”. *Journal of Neurology, Neurosurgery, and Psychiatry*. 68 (9): 405–15.
- [6] Flam, Faye (4 February 2016). “Fighting Zika Virus With Genetic Engineering”. *Bloomberg*.
- [7] S Huijben & KP Paaijmans. Putting evolution in elimination: winning our ongoing battle with evolving malaria mosquitoes and parasites. *Evolutionary Applications*, 2018, 11: 415-430
- [8] The malERA consultative group on resistance. Insecticide and drug resistance: The malERA Refresh research agenda for malaria elimination and eradication. *PLoS Medicine*, 2017, 14:e1002450
- [9] <https://store.arduino.cc/usa/duemk2>. “Especificaciones técnicas”.
- [10] <https://store.arduino.cc/usa/duemk2>. “Visión general”.
- [11] <https://store.arduino.cc/usa/duemk2>. “Documentación”.
- [12] Digital-output relative humidity & temperature sensor/module DHT22. Thomas Liu
- [13] <https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/>

- [14] https://naylampmechatronics.com/blog/15_Configuraci%C3%B3n-del-m%C3%B3dulo-bluetooth-HC-06-usa.html
- [15] Gupta, Ankur; Chakraborty, Saikat (January 2009). "Linear stability analysis of high- and low-dimensional models for describing mixing-limited pattern formation in homogeneous autocatalytic reactors". *Chemical Engineering Journal*.
- [16] H. Meinhardt, *Models of Biological Pattern Formation*, Academic Press (1982)
- [17] Murray, James D. (March 9, 2013). *Mathematical Biology*. Springer Science & Business Media. pp. 436–450. ISBN 978-3-662-08539-4.
- [18] Holmes, E. E.; Lewis, M. A.; Banks, J. E.; Veit, R. R. (1994). "Partial Differential Equations in Ecology: Spatial Interactions and Population Dynamics". *Ecology*. Wiley.
- [19] Murray, James D.; Stanley, E. A.; Brown, D. L. (November 22, 1986). "On the spatial spread of rabies among foxes". *Proceedings of the Royal Society of London. Series B. Biological Sciences*. The Royal Society.
- [20] Chaplan, M. A. J. (1995). "Reaction-difusión prepatterning and its potential role in tumour invasion". *Journal of Biological Systems*. World Scientific Pub Co Pte Lt.
- [21] Sherratt, J. A.; Nowak, M. A. (June 22, 1992). "Oncogenes, anti-oncogenes and the immune response to cancer : a mathematical model". *Proceedings of the Royal Society of London. Series B: Biological Sciences*. The Royal Society.
- [22] R.A. Gatenby and E.T. Gawlinski, *Cancer Res.* 56 (1996): 5745 Sherratt, J. A.; Murray, J. D. (July 23, 1990). "Models of epidermal wound healing". *Proceedings of the Royal Society of London. Series B: Biological Sciences*. The Royal Society.
- [23] P. Grindrod, *Patterns and Waves: The Theory and Applications of Reaction-Diffusion Equations*, Clarendon Press (1991)
- [24] J. Smoller, *Shock Waves and Reaction Diffusion Equations*, Springer (1994)
- [25] Maini PK, Baker RE, Chuong CM. 2006. Developmental biology. The Turing model comes of molecular age. *Science*. 2006 Dec 1; 314(5804): 1397–1398.
- [26] Joseph T. Belter, MS, BS, Jacob L. Segil, Aaron M. Dollar, PhD, SM, BS. Mechanical design and performance specifications of anthropomorphic prosthetic hands.
- [27] Predicción de las fuentes de reproducción de mosquitos mediante el uso de dispositivos electrónicos e inteligencia artificial. Eduardo Mellado Villaseñor. 25 de Junio de 2018. Tesis de Licenciatura, UMSNH (2018).
- [28] Application of an artificial neural network (ANN) model for predicting mosquito abundances in urban areas, Keun Young, Leea Namil, Chungb Suntae Hwangc.
- [29] Numerical Analysis - Burden & Faires - 9th Chapter 5, page 268.

- [30] Numerical Analysis - Burden & Faires - 9th Chapter 5, page 280. Theorem 5.12.
- [31] Fulks, W., Advanced calculus, (Third edition), John Wiley & Sons, New York, 1978, 731 pp. QA303.F954 10, 283
- [32] Butcher, J. C., The non-existence of ten-stage eighth-order explicit Runge-Kutta methods, BIT 25(1985), 521–542, QA76.N62 290.
- [33] Numerical Analysis - Burden & Faires - 9th Chapter 5, page 291. Tabla 5.10.
- [34] Numerical Analysis - Burden & Faires - 9th Chapter 12, page 713.
- [35] Numerical Analysis - Burden & Faires - 9th Chapter 12, page 714.
- [36] Numerical Analysis - Burden & Faires - 9th Chapter 12, page 715.
- [37] Numerical Analysis - Burden & Faires - 9th Chapter 12, page 742.
- [38] Miller, K. D. in Models of Neural Networks, III (eds. Domany, E., van Hemmen, J. L. & Schulten, K.) 55–78 (Springer, New York, 1996).
- [39] A logical calculus of the ideas immanent in nervous activity. Warren McCulloch. Article, The bulletin of mathematical biophysics volume 5, páginas 115–133(1943)
- [40] Hebb, D. O. The Organization of Behavior: A Neuropsychological Theory (Wiley, New York, 1949).
- [41] Chakravorty, Pragnan (2018). "What is a Signal? [Lecture Notes]". IEEE Signal Processing Magazine. 35 (5): 175–177. Bibcode:2018ISPM...35..175C. doi:10.1109/MSP.2018.2832195.
- [42] Gonzalez, Rafael (2018). Digital image processing. New York, NY: Pearson. ISBN 978-0-13-335672-4. OCLC 966609831.
- [43] Mosquitoes of Michigan - Their Biology and Control. Michigan Mosquito Control Organization. 2013.
- [44] The Deadliest Animal in the World. Gates, Bill. <https://www.gatesnotes.com/health/most-lethal-animal-mosquito-week>.
- [45] ISGLOBAL.org: Mosquito: el animal más letal del mundo.
- [46] Caracterización del MDF producido a partir de eucalipto y adhesivo. Cristiane Inácio de Campos; Francisco Antonio Rocco Lahr. Maderas: Ciencia y tecnología, ISSN 0717-3644, Vol. 6, N.º. 1, 2004, págs. 73-84
- [47] Society, National Geographic. "Mosquitoes, Mosquito Pictures, Mosquito Facts – National Geographic". National Geographic. Retrieved 2016-03-20.
- [48] The carnivores, Toxorhynchites . Wing Beats. Jones, C.; Schreiber, E. (1994).

- [49] Solución numérica de la ecuación reacción-difusión aplicada a tumores cerebrales, Tesis de la Lic. Venecia Chávez Medina, 2017, pag.2-7.
- [50] A logical calculus of the ideas immanent in nervous activity. Warren McCulloch.
- [51] Lecciones populares de matemáticas. Sistemas de numeración. S. V. Fomín. Editorial MIR. Cp. Origen de sistema de numeración.
- [52] Advanced Binary for Programming & Computer Science: Logical, Bitwise and Arithmetic Operations, and Data Encoding and Representation- 1st Edición, de Sunil Tanna (Author)
- [53] <https://en.wikipedia.org/wiki/Hexadecimal>
- [54] <https://www.microchip.com/mplab/avr-support/atmel-studio-7>
- [55] <http://www.asciitable.com/>