



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA



SIMULACIÓN
DOCUMENTACIÓN DEL PROYECTO DE SIMULACIÓN

Profesor:
MARTÍNEZ NIETO ADELINA

Integrantes de Equipo:
AGUILAR AMBROSIO SAMUEL
HERNANDEZ REYES PEDRO DE JESUS
RUIZ GUERRERO GABRIEL VALENTINO
VARELA VERA BIANI BISALUA

Grupo:
5SA

Grupo:
07:00 - 08:00 Hrs.

Contenido

RESUMEN	3
INTRODUCCIÓN	4
ETAPAS DEL MODELO DE SIMULACIÓN	5
Planteamiento del problema.....	5
Diseño del modelo	5
Modelo Ross-MacDonald	5
Variables del modelo.....	6
Estructura del modelo.....	8
Recolección de datos	10
PROTOTIPO	13
IMPLEMENTACIÓN DEL MODELO	15
VERIFICACIÓN	28
VALIDACIÓN	29
INTERPRETACIÓN DE RESULTADOS	31
CONCLUSIÓN	36
REFERENCIAS	40

RESUMEN

En el primer capítulo se analiza y explica el modelo de Ross- McDonald para predecir la cantidad de vectores (mosquitos) y personas contagiadas por Dengue, los objetivos de lo que va nuestro modelo de simulación, al igual que sus alcances y limitaciones de nuestro proyecto.

En el segundo capítulo, lo que se mostrará será conceptos importantes para poder ayudar al lector sobre cada uno de los temas relacionados a nuestra problemática y la metodología a elegir para nuestro modelo de simulación.

A continuación, en el tercer capítulo, con la metodología que elegimos, la empezamos a desglosar en nuestro modelo de simulación recabando datos, modelos y también la implementación de ecuaciones en lenguaje de programación.

Por último, se da la conclusión redactando el modelo de simulación que se utilizó, al igual las técnicas que se ocuparon, y lo más importante, los aprendizajes esperados al realizar el proyecto de nuestro modelo de simulación.

INTRODUCCIÓN

La simulación es una poderosa herramienta que permite recrear y estudiar sistemas complejos en un entorno virtual. A través de modelos matemáticos y computacionales, la simulación ofrece la posibilidad de analizar el comportamiento, rendimiento y características de diversos fenómenos y procesos, sin la necesidad de realizar experimentos reales o poner en riesgo recursos o vidas humanas.

Para llevar a cabo una simulación, es necesario desarrollar un modelo que describa las características esenciales del sistema en cuestión, como sus componentes, interacciones, reglas y restricciones. Estos modelos pueden ser de diferentes tipos, como modelos físicos, modelos matemáticos o modelos basados en agentes, dependiendo de la naturaleza del sistema y los objetivos de la simulación.

Una vez construido el modelo, se utiliza un software especializado de simulación para ejecutarlo y obtener resultados. Durante la simulación, se pueden modificar diferentes variables o condiciones del sistema para evaluar su impacto en los resultados y realizar análisis detallados. Esto permite explorar diferentes escenarios y tomar decisiones basadas en evidencia, lo que resulta invaluable en situaciones donde los experimentos reales son costosos, peligrosos o simplemente no son factibles.

El modelo de Ross-McDonald es ampliamente utilizado en epidemiología para predecir la propagación de enfermedades transmitidas por vectores, como el dengue.

Este modelo se basa en la interacción entre los vectores (mosquitos) y los hospedadores (humanos) para predecir la cantidad de personas contagiadas en un período de tiempo específico.

ETAPAS DEL MODELO DE SIMULACIÓN

Planteamiento del problema

El virus del Dengue es una enfermedad transmitida principalmente por mosquitos de la especie *Aedes*. Si bien la mayoría de las personas no presentan síntomas o solo tienen síntomas leves, la infección puede ser grave o mortal para los fetos de mujeres embarazadas. Dado que no hay vacunas ni tratamientos específicos para el virus del Dengue, la prevención es la principal estrategia de control, y una de las formas más efectivas de prevenir brotes es comprender y predecir la propagación de la enfermedad.

Por ello se usan simulaciones con datos históricos y actuales para predecir la propagación futura de la enfermedad. Y que pueda permitir a los responsables de la toma de decisiones prever dónde y cuándo pueden ocurrir brotes, y por lo tanto implementar medidas de control y prevención más efectivas. Además de que pueden ayudar a planificar la asignación de recursos, como insecticidas o mosquiteros tratados con insecticida, especialmente en áreas donde estos recursos son escasos.

Es así como se plantea realizar un modelo de simulación con el que, comprender e implementar mediante un programa y las fórmulas que se desarrollen como tal para el modelo, una interfaz en la que ingresando ciertos datos se obtenga una gráfica en la que se pueda observar el comportamiento de los contagios y con ello ver hacia donde se encaminaría.

Diseño del modelo

Modelo Ross-MacDonald

Este modelo aproxima el número de personas y vectores (*Aedes Aegypti*) infectados por el virus, mediante un sistema de dos ecuaciones diferenciales no lineales que retratan las interacciones entre personas y vectores, analizando mediante enfoque cuantitativo, alcance descriptivo y diseño no experimental trasversal. Consiste en la interacción de 4 poblaciones distribuidas de la siguiente manera: la población de humanos y la población del vector y estas dos a su vez se dividen en susceptibles de infectados, esto formando cuatro

grupos; humanos susceptibles, humanos infectados, mosquitos susceptibles y mosquitos infectados, además mediante el uso de ecuaciones diferenciales aproxima el número de personas infectadas y mosquitos infectados en un tiempo determinado, estas dos ecuaciones miden para la primera, la tasa (mediante una derivada) de cambio con respecto al tiempo que tienen las personas susceptibles al pasar, a ser infectadas y de igual forma para la segunda ecuación mide la tasa de cambio que tiene el vector susceptible a pasar a ser infectado y de esta forma aproxima el número de personas y mosquito que estarán infectados con el virus a lo largo del tiempo con las condiciones iniciales estipuladas. Además, las suposiciones del modelo son:

- Las poblaciones de humanos y mosquitos son homogéneas en cuanto a
- Susceptibilidad y exposición.
- Las poblaciones de mosquitos y humanos se mantienen constantes en el tiempo, es decir son poblaciones cerradas. Se ignoran los períodos de incubación dentro de los humanos y mosquitos.
- Se ignora la adquisición gradual de inmunidad en los humanos.
- La tasa per cápita (por cada uno) de recuperación de los humanos es mucho más alta que su tasa per cápita de mortalidad, justamente por eso se ignora la mortalidad de los humanos.
- Los mosquitos no se recuperan; no se ignora la mortalidad del vector.
- No se produce super infección en humanos o mosquitos.

Variables del modelo

Este como cualquier modelo requiere ciertos parámetros para poder estimar el comportamiento de una enfermedad, parámetros como la población de personas infectadas, el número de personas totales de la población o la probabilidad de transmisión de la enfermedad. Así que en este apartado se muestra y estima todas las variables, probabilidades y poblaciones que requiere el modelo, el cual está representado al alto índice de este tipo de enfermedades, se advierte que algunos parámetros son meramente

aproximaciones de las literaturas, ya que algunos parámetros son difíciles de calcular. La densidad de vector por persona “q” viene dada por la cantidad de mosquitos (vector) “M” entre la cantidad de personas en la población humana “N”, esto es $q=M/N$.

El *Aedes Aegypti* vive entre 20 a 30 días, es por esto que la tasa per cápita de mortalidad del vector (v) está entre $1/30 \leq v \leq 1/20$ por día, el período de tiempo entre dos comidas sanguíneas consecutivas se conoce como ciclo gonadotrófico (y), que en este caso se sugiere que $y = 3$ ya que, en promedio el *Aedes Aegypti* pica cada tres días; además h será al índice de antropofilia del vector, ya que se supone que el *Aedes Aegypti* se alimenta únicamente de sangre humana es decir, que es casi en su totalidad antropofílico por lo que el valor de “h” será 1 o muy cercano entonces $h \approx 1$, sin embargo para otras especies de mosquitos con índices de antropofilia h estaría entre $0 \leq h \leq 1$, siendo 0 un vector no antropofílico y 1 en su totalidad antropofílico.

Es por eso por lo que la frecuencia de picada sobre humano (que identificaremos como a) estará dada por $a = h/y$; sustituyendo los datos obtenidos tenemos que esta será $a \approx 1/3$.

La probabilidad de que una picada origine una infección en el vector “c” es de $c=0.08058$, de igual forma la probabilidad que una picada genere una infección en el humano (p), la cual estará dado por $p = 0.22687$, estas dos probabilidades suelen variar de $0.03 \leq c \leq 0.08$ y $0.1 \leq p \leq 0.3$.

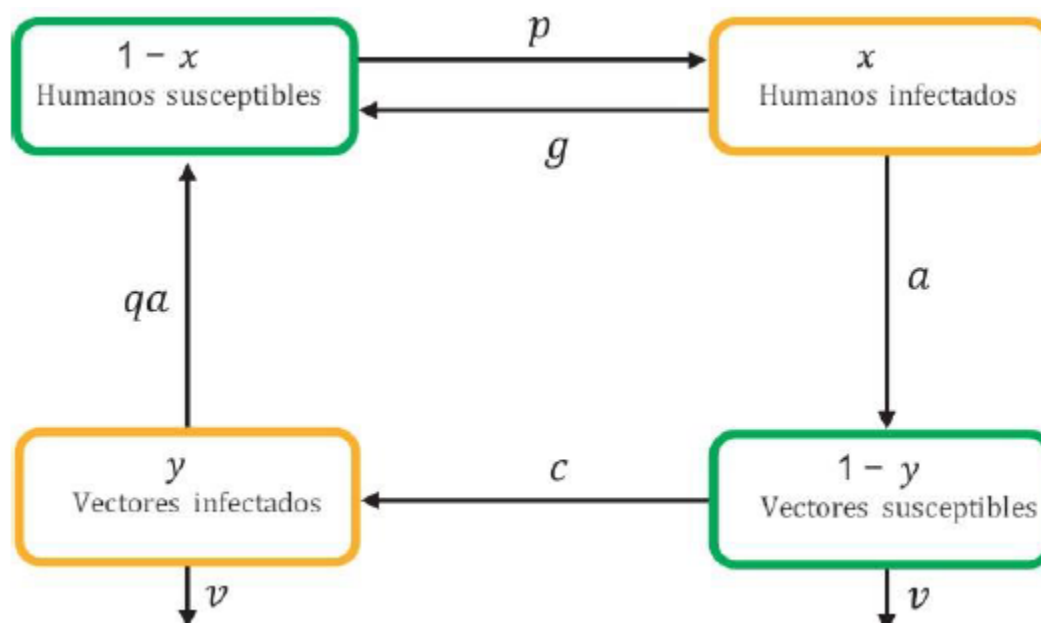
Y, además, para que haya una epidemia es necesario que hubiera una primera infección ya sea de un mosquito o un humano, a este elemento se le llama condición inicial “X0 ” y representa el número de personas infectadas que llegaron a una población de susceptibles, también existe condición inicial para el mosquito “Y0 ”, aunque esta usualmente se toma como $Y_0=0$, ya que se le da más importancia a la movilidad de las personas al ser mayor que la del mosquito, además de que esta es difícil de calcular y solo se puede suponer.

Tabla No. 1: Significado de variables

Parámetro	Significado
N	Población de humanos.
M	Población de vectores.
$q = \frac{M}{N}$	Densidad de vectores por persona.
h	Índice de antropofilia.
γ	Duración de ciclo gonadotrófico.
$a = \frac{h}{\gamma}$	Tasa de picadura por mosquito.
p	Probabilidad de que una picadura genere una infección en humanos.
c	Probabilidad de que una picadura genere una infección en vectores.
g	Tasa de recuperación del humano.
v	Tasa de mortalidad del vector.
x	Proporción de personas infectadas.
y	Proporción de vectores infectados.
X	Número absoluto de personas infectadas.
Y	Número absoluto de vectores infectados.

Estructura del modelo

Figura No. 1: Interacción entre humanos y vectores(mosquitos).



Nota: como se puede ver en la Figura No. 1, se muestra las diferentes interacciones entre humanos y los vectores (mosquitos).

Primero se puede notar que, en la interacción de personas susceptibles a infectadas, las personas pasan de susceptibles a infectadas por medio de “p”, pasan de infectadas a susceptibles por medio de “g”. luego las personas infectadas infectan a los mosquitos susceptibles de acuerdo con “a”, se observa que a partir de este punto puede haber una salida del mosquitos del sistema debido a la mortalidad de mosquitos susceptibles y también de mosquitos infectados, lo que representamos con “v”, los mosquitos pasan a ser infectados por medio de la probabilidad “c”, y para cerrar el ciclo, los mosquitos infectan a los humanos susceptibles por medio de la multiplicación “qa” esto debido a que “q” es la densidad del mosquito es decir el número de mosquitos por personas, ya que habrá “qa” mosquitos listos para picar a cada persona.

Es así que el modelo tendrá la siguiente forma:

$$Dx/dt = qapy(1 - x) - gx \quad (1)$$

$$Dy/dt = acx(1 - y) - vy \quad (2)$$

En (1) se establece que la variación de personas enfermas de Dengue, será directamente proporcional al número de personas susceptibles, incluidas las que ya padecieron la enfermedad, la densidad de mosquitos por persona y los mosquitos infectados, es decir el número de personas susceptibles $(1 - x)$ se infectan mediante la interacción de los vectores infectados “y” los cuales tienen una probabilidad “p” de infectar a dichas personas y todo esto multiplicando por la densidad del vector “q”, de lo cual surge la expresión $qapy(1 - x)$, además las personas infectadas “x” se recuperan a razón de “g” y pasan a ser susceptibles nuevamente es decir la expresión “xg”, este cambio de población se representa mediante la diferencia de dichas expresiones es decir $qapy(1 - x) - xg$, y como se dijo, al ser directamente proporcional a la variación de las personas infectadas surge la ecuación diferencial $\dot{x} = qapy(1 - x) - xg$.

Para (2) La variación de mosquitos infectados será directamente proporcional al número de mosquitos susceptibles, al número de personas infectadas y la tasa de

mortalidad de mosquitos, o bien del mismo modo los vectores susceptibles “1-y” se infectan con una probabilidad “c” cuando se alimentan sobre humanos infectados “x” con una tasa de picadura por mosquito sobre humano “a”, de esto surge la expresión $acx(1 - y)$ la cual representa la proporción de vectores que pasan a ser infectados, y los vectores infectados mueren a una tasa per cápita de mortalidad “v” es decir la expresión “vy”, y la diferencia de las dos expresiones representa el número de vectores infectados respecto al tiempo, es decir la ecuación diferencial $\dot{y}=acx(1 - y)-vy$.

Pero además se modificó el modelo para que este sea un modelo de valor absoluto, ya que era porcentual, dicha modificación ayudó a convertir el resultado arrojado en números absolutos no relativos, es decir, el número total de infectados, para ello decimos que el número absoluto de personas infectadas “x” fue “xN”, siendo “N” la población de las personas, y lo llamaremos $xN = X$, de igual forma para los vectores, diremos que el número absoluto no relativo de vectores infectados fue $yM = Y$, teniendo esto en cuenta, el sistema de ecuaciones (1) y (2) se modificó de la forma siguiente: multiplicaremos por “N” a ambos lados de la ecuación (1), lo cual según las reglas básicas de matemáticas no modificó en lo absoluto la ecuación, y al multiplicar “M” a ambos lados de (2), además recordando que $q=M/N$, entonces se obtiene el sistema:

$$\begin{aligned} \frac{dX}{dt} &= (a/N * p) (N - X) Y - gX \\ \frac{dY}{dt} &= (a/N * c) (M - Y) X - vY \end{aligned}$$



Recolección de datos

Para realizar la colección de datos, se consultaron diversas páginas de internet, en donde logramos obtener algunos datos que necesitamos, y los demás los tomaremos del autor debido a que son difíciles de calcular y ya se realizaron estudios que comprueban que dichas variables se encuentran en los rangos que se dieron anteriormente.

Síntomas

En la mayor parte de los casos, el dengue causa síntomas leves o incluso ningún síntoma y se cura en una o dos semanas, pero en casos infrecuentes se agrava y puede causar la muerte.

Cuando aparecen síntomas, lo suelen hacer entre 4 y 10 días después de la infección y duran de 2 a 7 días. Pueden ser:

Ciclo de vida	
	El mosquito puede completar su ciclo de vida, desde el huevo hasta el adulto, en 7-10 días; los mosquitos adultos generalmente viven de 4 a 6 semanas. La hembra <i>Aedes aegypti</i> es responsable de la transmisión de enfermedades porque necesita sangre humana para el desarrollo de sus óvulos y para su metabolismo. El macho no se alimenta de sangre.
Actividad	
	El mosquito es más activo temprano en la mañana y al anochecer, por lo que estos son los periodos de mayor riesgo de picaduras. Sin embargo, las hembras, que necesitan continuar alimentándose, buscarán una fuente de sangre en otros momentos. La hembra <i>Aedes aegypti</i> se alimenta cada 3-4 días; sin embargo, si no pueden extraer suficiente sangre, continúan alimentándose cada momento que pueden.

Uno de los principales transmisores del dengue es el *Aedes aegypti*, a quien tomamos como vector en esta simulación, en la imagen anterior se muestran los datos para la tasa de mortalidad del mosquito, que vive de 4 a 6 semanas, podemos tomar en este rango que la tasa de mortalidad va de $\frac{1}{30} \leq \nu \leq \frac{1}{20}$.

Igualmente nos menciona el artículo, que este mosquito se alimenta cada 3-4 días, podemos tomar que la tasa de picadura por mosquito es de $y = \frac{1}{3}$.



En esta simulación tomamos como ubicación Oaxaca y de esta ficha podemos extraer el dato de la población humana que es de 3967889.

Dado que no hay un registro de la población de vectores, este dato se calcula en base a la población humana, tomamos como punto que la densidad de vectores por persona sea de 4, por lo que la población de vectores es de 15871556.

PROTOTIPO

Se elaboró un prototipo de cómo se visualizarían cada una de las pantallas al usuario, en esta primera imagen se visualiza la pantalla de inicio, en donde se le da al usuario una pequeña información acerca de que trata el proyecto y una explicación en que consiste el modelo Ross-Macdonald, cuando el usuario quiera ir a la simulación, solo basta con darle click al botón ir a la simulación que lo redireccionará a la pantalla en donde podrá hacer la simulación.

SOFTWARE DE SIMULACIÓN - VECTORES (Mosquitos) Y PERSONAS CONTAGIADAS POR DENGUE

¡BIENVENIDO!

ACERCA DE ESTE PROYECTO:

Pronostica la dinamica de transmision del virus del dengue en el estado de Oaxaca a traves del modelo matematico Ross-Macdonald.

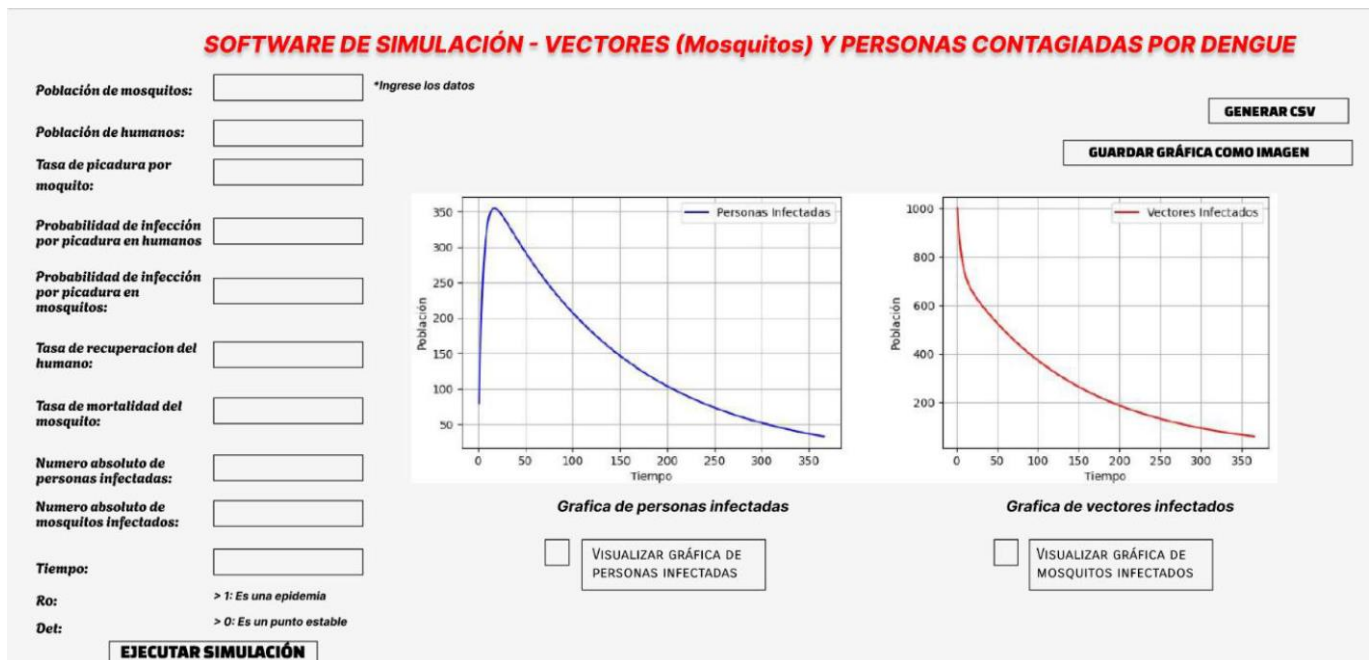
Se le proporcionara resultados precisos sobre el número de personas y mosquitos infectados, así como resultados sobre cómo es la evolución de la epidemia en el estado.

IR A LA SIMULACIÓN

MODELO ROSS-MACDONALD:

Este modelo consta de dos ecuaciones diferenciales y consiste en la interacción de 4 poblaciones distribuidas de la siguiente manera: la población de humanos y la población del mosquito y estas dos a su vez se dividen en susceptibles e infectados, esto formando cuatro grupos: humanos susceptibles, humanos infectados, mosquitos susceptibles y mosquitos infectados, además mediante el uso de ecuaciones diferenciales, aproxima el número de personas infectadas y mosquitos infectados en un tiempo determinado, estas dos ecuaciones miden, para la primera la tasa (mediante una derivada) de cambio con respecto al tiempo que tienen las personas susceptibles al pasar a ser infectadas y de igual forma para la segunda ecuación mide la tasa de cambio que tiene el vector susceptible a pasar a ser infectado y de esta forma, aproximó el número de personas y mosquito que estaban infectados con el virus a lo largo del tiempo con las condiciones iniciales estipuladas.

En esta imagen, se visualiza como sería la pantalla de simulación, en donde cuenta con un título en la parte superior de la pantalla, del lado izquierdo se encuentran todos los parámetros que se requieren para la simulación y donde el usuario tendrá que ir ingresando cada uno de los datos para poder hacer la simulación, así como también, debajo de todos los datos se encuentra el botón iniciar simulación, en donde con ese botón dándole click podrá iniciar la simulación y se mostraran del lado derecho las gráficas de propagación del dengue, se mostraran 2 graficas, en donde una primer grafica corresponde a las personas infectadas y la segunda grafica a los vectores(mosquitos) infectados, el usuario podrá elegir con ayuda de un check list cuál de las 2 graficas quiere visualizar o las 2 graficas, también podrá guardar las o la gráfica como imagen y guardar todos los datos en un archivo csv.



IMPLEMENTACIÓN DEL MODELO

Se presenta a continuación unas ilustraciones de las ecuaciones de lenguaje de programación que ocupamos para nuestro modelo de simulación, con una breve descripción de su función.

```
def RossMacdonald(y, t, a, c, M, N, p, g, v):  
    P, V = y  
    dPdt = (((a/N)*p)*(N-P)*V)-(g*P)  
    dVdt = (((a/N)*c)*(M-V)*P)-(v*V)  
    return [dPdt, dVdt]
```

En este fragmento de código, se declara el modelo matemático de Ross Macdonald y se le pasan los parámetros de entrada que requiere para su ejecución, así como también se calcula la tasa de cambio de la población humana en el tiempo (dPdt) y la tasa de cambio de la población de vectores (mosquitos) en el tiempo (dVdt) y la función devolverá la lista con los dos elementos que se calcularon.

```
def calcular_R0(a, c, M, N, p, g, v):  
    return ((M/N) * (a**2) * c * p) / (v * g)
```

Con esta función que se declaró como cacular_R0, nos va a calcular el número básico de reproducción, que va a indicar cuantas personas puede infectar un individuo infectado en una población que se considere susceptible.

```
def calcular_Det(a, c, M, N, p, g, v):  
    return (1/N) * (((a**2) * c * p * M) - (N * v * g))
```

Con esta función que se declaró como calcular_Det, va a medir el equilibrio entre la transmisión de la enfermedad y la mortalidad de humanos y vectores.

```
def plot_results():
```

Con esta función, se generará la gráfica donde se van a visualizar los resultados.

```
class AplicacionSimulacionDengue:
    def __init__(self, raiz):
        self.raiz = raiz
        self.raiz.title("Software de Simulación de Dengue")
        self.raiz.geometry("1366x769")

        self.crear_pantalla_bienvenida()
```

Se crea una interfaz gráfica, que es la página de inicio de la simulación, se configura la ventana con un título y tamaño en especial, con ayuda de un método llamado crear_pantalla_bienvenida que es el encargado de mostrar la pantalla inicial con información que el usuario podrá visualizar.

```
def crear_pantalla_bienvenida(self):
    # Limpiar widgets existentes
    for widget in self.raiz.winfo_children():
        widget.destroy()
```

En este método, inicialmente se eliminan todos los widget existentes en la ventana principal para dejarla vacía antes de mostrar nuevos elementos, para así asegurar que cada pantalla se limpie.


```

titulo = tk.Label(
    self.raiz,
    text="SOFTWARE DE SIMULACIÓN - VECTORES (Mosquitos) Y PERSONAS CONTAGIADAS POR DENGUE",
    font=("Arial", 16, "bold"),
    fg="red"
)
titulo.pack(pady=10)

```

Se crea un título que sea visible en la pantalla principal, se especifican las medidas, el texto y color del texto.

```

bienvenido = tk.Label(self.raiz, text="¡BIENVENIDO!", font=("Arial", 36, "bold"), fg="red")
bienvenido.pack(pady=20)

```

Se crea el mensaje de bienvenida que tendrá la pantalla inicial, se ajustan las medidas del mensaje que el usuario visualizara.

```

frame_textos = tk.Frame(self.raiz)
frame_textos.pack(pady=20, padx=20, fill="both", expand=True)

```

Se crea un frame dentro de la pantalla inicial, esto se hace para poder expandirse y ocupar todo el espacio disponible.

```

texto1_titulo = tk.Label(
    frame_textos,
    text="ACERCA DE ESTE PROYECTO:",
    font=("Arial", 14, "bold"),
    anchor="w"
)
texto1_titulo.grid(row=0, column=0, sticky="w", padx=20, pady=10)

```

Se crea una etiqueta, se añade el texto que contendrá la etiqueta y se ajustan las medidas, tanto del texto como de la etiqueta.

```

texto1 = tk.Label(
    frame_textos,
    text=(
        "Pronostica la dinámica de transmisión del virus del dengue en el estado "
        "de Oaxaca a través del modelo matemático Ross-Macdonald.\n\n"
        "Se le proporcionará resultados precisos sobre el número de personas y "
        "mosquitos infectados, así como resultados sobre cómo es la evolución de "
        "la epidemia en el estado."
    ),
    font=("Arial", 12),
    justify="left",
    wraplength=600
)
texto1.grid(row=1, column=0, sticky="w", padx=20)

```

Se crea otra etiqueta en donde contendrá un texto descriptivo sobre el proyecto, se añade el texto y se ajustan las medidas.

```

texto2_titulo = tk.Label(
    frame_textos,
    text="MODELO ROSS-MACDONALD:",
    font=("Arial", 14, "bold"),
    anchor="w"
)
texto2_titulo.grid(row=0, column=1, sticky="w", padx=20, pady=10)

```

Se crea una etiqueta, se añade el texto que contendrá la etiqueta y se ajustan las medidas, tanto del texto como de la etiqueta.

```

texto2 = tk.Label(
    frame_textos,
    text=(
        "Este modelo consta de dos ecuaciones diferenciales y consiste en la interacción "
        "de 4 poblaciones distribuidas de la siguiente manera: la población de humanos y "
        "la población de mosquito y estas dos a su vez se dividen en susceptibles e infectados, "
        "esto formando cuatro grupos...\n\n"
        "Además mediante el uso de ecuaciones diferenciales, aproxima el número de personas infectadas "
        "y mosquitos infectados en un tiempo determinado..."
    ),
    font=("Arial", 12),
    justify="left",
    wraplength=600
)
texto2.grid(row=1, column=1, sticky="w", padx=20)

```

Se crea otra etiqueta en donde contendrá un texto descriptivo sobre el modelo Ross-Macdonald, se añade el texto y se ajustan las medidas.

```
frame_textos.columnconfigure(0, weight=1)
frame_textos.columnconfigure(1, weight=1)
```

Se ajustan las columnas del frame para que se ajusten al tamaño de la pantalla, si el tamaño de la pantalla cambia, ambas columnas se ajustaran de manera proporcional al espacio.

```
btn_simulacion = tk.Button(
    self.raiz,
    text="IR A LA SIMULACIÓN",
    font=("Arial", 14, "bold"),
    bg="white",
    fg="black",
    command=self.abrir_simulacion,
    width=40,
    height=2
)
btn_simulacion.pack(pady=150)
```

Se crea un botón que, al momento de ser presionado, se ejecutara la función de abrir_simulacion, este botón una vez presionado, va a redirigir al usuario a la pantalla de la simulación, se ajustan las medidas del botón.

```
def abrir_simulacion(self):
    for widget in self.raiz.winfo_children():
        widget.destroy()
```

Se limpia la ventana de cualquier widget visible antes de mostrar la simulación, con esto se pretende eliminar la pantalla de inicio y solo cargar los elementos gráficos de la pantalla de la simulación.

```
self.raiz.title("Software de Simulación")
canvas = None
canvas2 = None
```

Se establece el título que tendrá la pantalla de simulación y se definen dos variables que se inicializaran como none.

```
ancho_ventana = 1366
alto_ventana = 768
ancho_pantalla = self.raiz.winfo_screenwidth()
alto_pantalla = self.raiz.winfo_screenheight()
posicion_superior = int(alto_pantalla / 2 - alto_ventana / 2)
posicion_derecha = int(ancho_pantalla / 2 - ancho_ventana / 2)
self.raiz.geometry(f"{ancho_ventana}x{alto_ventana}+{posicion_derecha}+{posicion_superior}")
```

Se ajustan las dimensiones y posición de la pantalla, asegurando que la pantalla quede centrada.

```
frame_controles = tk.Frame(self.raiz)
frame_controles.pack(side=tk.LEFT, padx=20, pady=20)
```

Se crea un frame que es un contenedor que va a permitir agrupar otros widgets, se coloca dentro de la pantalla principal.

```
frame_centro_superior = tk.Frame(self.raiz, width=ancho_ventana, height=alto_ventana // 2)
frame_centro_superior.pack()
```

Se crea otro frame y se coloca dentro de la pantalla principal, permitiendo definir un área dentro de la ventana con un tamaño específico y se agrega a la interfaz.

```
frame_grafica = tk.Frame(self.raiz)
frame_grafica.pack(side=tk.RIGHT, padx=20, pady=20)
```

Se crea un frame que se colocara en el lado derecho de la pantalla principal con este frame se mostraran las gráficas.

```

etiquetas = [
    ("Poblacion\nde Mosquitos:", 0),
    ("Poblacion\nde Personas:", 1),
    ("Tasa de\nPicadura\npor Mosquito:", 2),
    ("Probabilidad de\nInfección\npor Picadura\nen Humanos:", 3),
    ("Probailidad de\nInfección\nPor Picadura\nen Mosquitos:", 4),
    ("Tasa de\nRecuperacion\nde Personas:", 5),
    ("Tasa de\nMortalidad\ndel mosquito:", 6),
    ("Numero absoluto\nde personas\ninfectadas:", 7),
    ("Numero absoluto\nde mosquitos\ninfectadas:", 8),
    ("Tiempo-dias:", 9)]
]
for texto, fila in etiquetas:
    tk.Label(frame_controles, text=texto).grid(row=fila, column=0, sticky="e")

```

Se genera un conjunto de etiquetas que describen los parámetros que son requeridos y necesarios para la simulación, cada etiqueta se coloca en una fila específica dentro del frame_controles para organizarlos en la interfaz y hacer que tengan una mejor visualización para el usuario.

```

entradas = {}
nombres_entradas = ['M', 'N', 'a', 'p', 'c', 'g', 'v', 'P0', 'V0']
for i, nombre in enumerate(nombres_entradas):
    entradas[nombre] = tk.Entry(frame_controles)
    entradas[nombre].grid(row=i, column=1)

```

Se crea un conjunto de entradas de texto que se usan para que el usuario vaya ingresando los valores que están asociados a los diferentes parámetros que se requieren para hacer la simulación, cada entrada se asocia con un nombre y se organiza en una fila dentro del frame_controles.

```

escala_tiempo = tk.Scale(frame_controles, from_=0, to=365, resolution=1, orient=tk.HORIZONTAL)
escala_tiempo.grid(row=9, column=1)

```

Se crea un control deslizante que le va a permitir al usuario seleccionar un valor de tiempo dentro de un rango ya determinado, en este caso el valor del tiempo se representa en días.

```

mensaje_R0 = tk.Label(frame_controles, text="")
mensaje_R0.grid(row=12, column=1)
mensaje_Det = tk.Label(frame_controles, text="")
mensaje_Det.grid(row=14, column=1)

```

Se crean 2 etiquetas vacías dentro del marco de controles para mostrar los mensajes del potencial de transmisión y el punto de equilibrio, estas etiquetas se irán actualizando de manera dinámica según sean los resultados que se obtengan.

```

valores_predeterminados = {
    'M': "15871556", 'N': "3967889", 'a': "0.33333", 'p': "0.22687",
    'c': "0.08058", 'g': "0.14285", 'v': "0.06666",
    'P0': "80", 'V0': "1000"
}
for nombre, valor in valores_predeterminados.items():
    entradas[nombre].insert(0, valor)
escala_tiempo.set(365)

```

Se establecen los valores predeterminados en las entradas de texto de la interfaz para los diferentes parámetros de la simulación, además se ajustan el control deslizante del tiempo en un valor máximo de 365 días.

```

def graficar_resultados():
    nonlocal canvas, canvas2

```

Se dibujan los gráficos del resultado de la simulación.

```

try:
    parametros = {nombre: float(entradas[nombre].get()) for nombre in nombres_entradas}
    tiempo = float(escala_tiempo.get())
except ValueError:
    messagebox.showerror("Error", "Por favor, ingrese valores numéricos válidos")
    return

```

Se obtiene los valores de entrada de texto y del control deslizante y los convierte a números de tipo float, si el usuario ingresa un tipo de datos no valido se le mostrara un mensaje de texto indicándole cual es el error y se detiene la ejecución.

```

t = np.linspace(1, tiempo, 365)
y0 = [parametros['P0'], parametros['V0']]
sol = odeint(RossMacdonald, y0, t, args=(
    parametros['a'], parametros['c'], parametros['M'], parametros['N'],
    parametros['p'], parametros['g'], parametros['v']
))

```

Se resuelve el sistema de ecuaciones diferenciales que modela la propagación del dengue con ayuda del modelo Ross-Macdonald, los resultados obtenidos se guardan en la variable sol.

```

if canvas is not None:
    canvas.get_tk_widget().destroy()
if canvas2 is not None:
    canvas2.get_tk_widget().destroy()

```

Se verifica si los objetos canva y canva2 existen, si se cumplen con la condición los elimina de la interfaz permitiendo así, eliminar los gráficos anteriores antes de mostrar los nuevos gráficos.

```

R0 = calcular_R0(**{k: parametros[k] for k in ['a', 'c', 'M', 'N', 'p', 'g', 'v']})
if R0 > 1:
    mensaje_R0.config(text="> 1: Es una epidemia")
elif R0 < 1:
    mensaje_R0.config(text="< 1: Tendrá un impacto menor")
else:
    mensaje_R0.config(text="= 1: Posibilidad de alcanzar indicadores endémicos o no")

```

En esta parte, se calcula el valor de R_0 con ayuda de la función `calcular_R0`, una vez obtenido el valor se muestra un mensaje en la interfaz gráfica dependiendo del rango que tome, si R_0 es mayor que 1 significa que la enfermedad puede propagarse y se actualiza el texto para mostrarse en la interfaz gráfica, si R_0 es menor a 1 significa que la enfermedad no puede propagarse y se actualiza el texto para mostrarse en la interfaz gráfica, si no se cumple con ninguno de los 2, entonces se salta al else y quiere decir que R_0 es igual 1 y significa que la enfermedad está en equilibrio y se actualiza el texto para mostrarse en la interfaz gráfica.

```

Det = calcular_Det(**{k: parametros[k] for k in ['a', 'c', 'M', 'N', 'p', 'g', 'v']})
if Det > 0:
    mensaje_Det.config(text="> 0: Es un punto estable")
else:
    mensaje_Det.config(text="< 0: Es un punto silla")

```

En esta parte, se calcula el valor e Det con ayuda de la función calcular_Det(), si det es mayor a 0 significa que el sistema se encuentra en un punto estable y se actualiza el texto para visualizarse en la interfaz gráfica, en caso contrario de que det sea menor a 0 significa que el sistema no se encuentra en un punto estable y se actualiza el texto para visualizarse en la interfaz gráfica.

```

fig = Figure(figsize=(6, 4), dpi=100)
ax = fig.add_subplot(111)
ax.plot(t, sol[:, 0], 'b', label='Personas Infectadas')
ax.legend(loc='best')
ax.set_xlabel('Tiempo')
ax.set_ylabel('Población')
ax.grid()

fig2 = Figure(figsize=(6, 4), dpi=100)
ax2 = fig2.add_subplot(111)
ax2.plot(t, sol[:, 1], 'r', label='Vectores Infectados')
ax2.legend(loc='best')
ax2.set_xlabel('Tiempo')
ax2.set_ylabel('Población')
ax2.grid()

```

En este pedazo de código, se crean las 2 graficas que el usuario va a visualizar, la primera grafica corresponde a la gráfica de personas infectadas con las leyendas correspondientes, la segunda grafica corresponde a la gráfica de vectores (mosquitos) infectados con las leyendas correspondientes para que el usuario pueda comprender y visualizar las gráficas.


```

canvas = FigureCanvasTkAgg(fig, master=frame_grafica)
canvas.draw()
canvas.get_tk_widget().pack(side=tk.LEFT)

canvas2 = FigureCanvasTkAgg(fig2, master=frame_grafica)
canvas2.draw()
canvas2.get_tk_widget().pack(side=tk.LEFT)

```

Aquí, se muestran las gráficas generadas anteriormente en la ventana de la interfaz gráfica.

```

boton_graficar = tk.Button(frame_controles, text="Ejecutar simulacion", command=plot_results)
boton_graficar.grid(row=15, column=1, pady=10)

```

Se crea un botón que va a permitir iniciar y ejecutar la simulación para así mostrar los resultados obtenidos en la interfaz gráfica.

```

raiz = tk.Tk()
aplicacion = AplicacionSimulacionDengue(raiz)
raiz.mainloop()

```

Para generar el documento pdf se realizó una función para disponer que el usuario guarde en el sitio de su preferencia el documento

```

def generar_pdf(self, fig1, fig2, anos_seleccionados, parametros_entrada, parametros_historicos):
    # Abrir ventana para seleccionar dónde guardar el archivo
    ruta_pdf = filedialog.asksaveasfilename(defaultextension=".pdf")
    if not ruta_pdf: # Si el usuario cancela o no selecciona una ruta, salir de la función
        messagebox.showinfo("Operación cancelada", "No se seleccionó una ubicación para guardar el PDF.")
        return

```

Posterior a eso empezaremos a obtener la información del proyecto a el pdf obteniendo primero las imágenes

```

181
182     # Crear un directorio temporal para guardar las imágenes
183     with tempfile.TemporaryDirectory() as tmpdirname:
184         # Guardar las figuras como imágenes PNG
185         ruta_grafica1 = os.path.join(tmpdirname, 'grafica1.png')
186         ruta_grafica2 = os.path.join(tmpdirname, 'grafica2.png')
187
188         fig1.savefig(ruta_grafica1, bbox_inches='tight') # Guardar la primera figura
189         fig2.savefig(ruta_grafica2, bbox_inches='tight') # Guardar la segunda figura
190

```

Pero para ello se necesita crear el pdf con su titulo

```
# Crear el documento PDF
doc = SimpleDocTemplate(ruta_pdf, pagesize=letter)
elementos = []
estilos = getSampleStyleSheet()

# Título
titulo = Paragraph("Simulación de Dengue - Modelo Ross-Macdonald en Oaxaca", estilos['Title'])
elementos.append(titulo)
elementos.append(Spacer(1, 12))
```

Empezamos a obtener los datos e irlos guardando en una tabla para tener un orden en mostrar los resultados

```
# Datos de los años seleccionados
datos_tabla = [{"Año", "Poblacion\nde\nmosquitos", "Poblacion\nde\nPersonas", "Tasa de\npicadura\npor Mosquito", "Probabilidad\nde\ninfeccion por\nPica"}]
for ano in anos seleccionados:
    datos = parametros historicos.get(ano, {})
    datos_tabla.append([
        ano,
        datos.get('M', 'N/A'), datos.get('N', 'N/A'), datos.get('a', 'N/A'), datos.get('p', 'N/A'),
        datos.get('c', 'N/A'), datos.get('g', 'N/A'), datos.get('v', 'N/A'),
        datos.get('P0', 'N/A'), datos.get('v0', 'N/A')
    ])

tabla_anos = Table(datos_tabla)
tabla_anos.setStyle(TableStyle([
    ('BACKGROUND', (0,0), (-1,0), colors.blue),
    ('TEXTCOLOR', (0,0), (-1,0), colors.whitesmoke),
    ('ALIGN', (0,0), (-1,-1), 'CENTER'),
    ('FONTNAME', (0,0), (-1,0), 'Helvetica-Bold'),
    ('FONTSIZE', (0,0), (-1,0), 7),
    ('BOTTOMPADDING', (0,0), (-1,0), 12),
    ('BACKGROUND', (0,1), (-1,-1), colors.beige),
    ('GRID', (0,0), (-1,-1), 1, colors.black)
]))
elementos.append(Paragraph("Datos del Programa Epidemiologico:", estilos['Heading3']))
elementos.append(tabla_anos)
elementos.append(Spacer(1, 12))
```

De igual manera empezamos a obtener los datos que se introdujeron en la simulación i se van agregando otra tabla

```

# Datos de simulación
datos_simulacion = [
    ["Parámetro", "Valor"],
    ["Poblacion de\nMosquitos", parametros_entrada.get('M', 'N/A')],
    ["Poblacion de\nPersonas", parametros_entrada.get('N', 'N/A')],
    ["Tasa de picadura\npor Mosquitos", parametros_entrada.get('a', 'N/A')],
    ["Probabilidad de\ninfeccion por \nPicadura en Humanos", parametros_entrada.get('p', 'N/A')],
    ["Probabilidad de\ninfeccion por\nPicadura en Mosquitos", parametros_entrada.get('c', 'N/A')],
    ["Tasa de Recuperacion\nen Personas", parametros_entrada.get('g', 'N/A')],
    ["v", parametros_entrada.get('v', 'N/A')],
    ["P0", parametros_entrada.get('P0', 'N/A')],
    ["V0", parametros_entrada.get('V0', 'N/A')]
]

tabla_simulacion = Table(datos_simulacion)
tabla_simulacion.setStyle(TableStyle([
    ('BACKGROUND', (0,0), (-1,0), colors.blue),
    ('TEXTCOLOR', (0,0), (-1,0), colors.whitesmoke),
    ('ALIGN', (0,0), (-1,-1), 'CENTER'),
    ('FONTNAME', (0,0), (-1,0), 'Helvetica-Bold'),
    ('FONTSIZE', (0,0), (-1,0), 12),
    ('BOTTOMPADDING', (0,0), (-1,0), 12),
    ('BACKGROUND', (0,1), (-1,-1), colors.beige),
    ('GRID', (0,0), (-1,-1), 1, colors.black)
]))

elementos.append(Paragraph("Datos de Simulación:", estilos['Heading3']))
elementos.append(tabla_simulacion)
elementos.append(Spacer(1, 12))

```

Por ultimo se añaden las imágenes al pdf y se construye

```

# Añadir las gráficas al PDF
elementos.append(Image(ruta_grafica1)) # Agregar la primera gráfica
elementos.append(Paragraph("Gráfica de Personas Infectadas:", estilos['Heading3']))

elementos.append(Image(ruta_grafica2)) # Agregar la segunda gráfica
elementos.append(Paragraph("Gráfica de Mosquitos Infectados:", estilos['Heading3']))

# Construir el PDF
doc.build(elementos)

# Mostrar mensaje de éxito
messagebox.showinfo("PDF Generado", f"El archivo se guardó en: {ruta_pdf}")

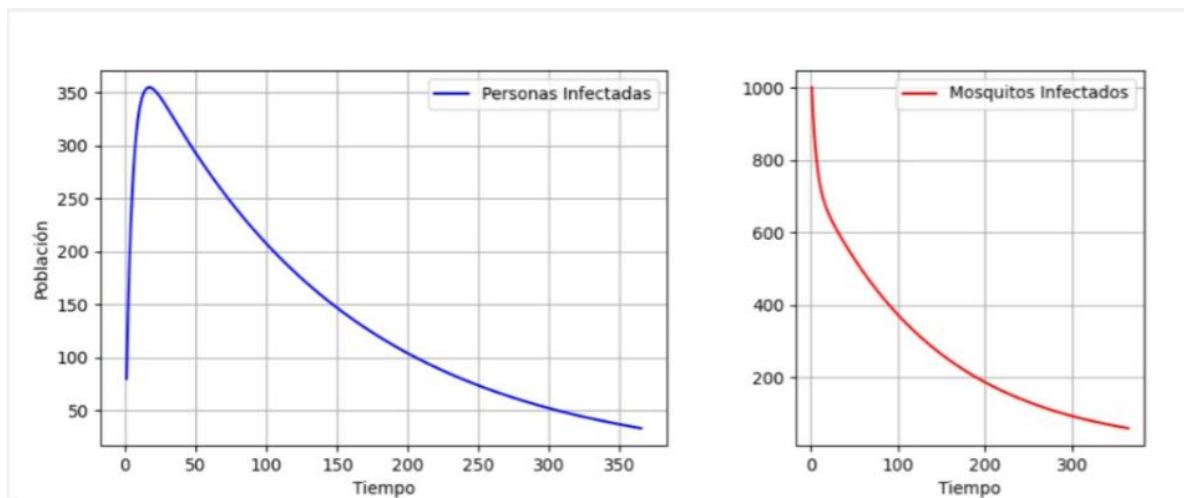
```

VERIFICACIÓN

El modelo funciona según lo esperado

Poblacion mosquitos:	15871556
Poblacion humanos:	3967889
Tasa de picadura por mosquito:	0.33333
Probabilidad de infeccion por picadura en humanos:	0.22687
Probabilidad de infeccion por picadura en mosquito:	0.08058
Tasa de recuperacion del humano:	0.14285
Tasa de de mortalidad del mosquito:	0.06666
Numero absoluto de personas infectadas:	80
Numero absoluto de mosquitos infectados:	1000
Tiempo-Dias:	365
Potencial de trasmision:	
Punto de equilibrio:	
<input type="button" value="Ejecutar simulacion"/>	

Desde la inserción dinámica de datos y poder actualizar las gráficas.



Las mismas gráficas se comportan según lo esperado tomando en cuenta los ejes para la población de humanos, vectores y los días estimados para la simulación.

VALIDACIÓN

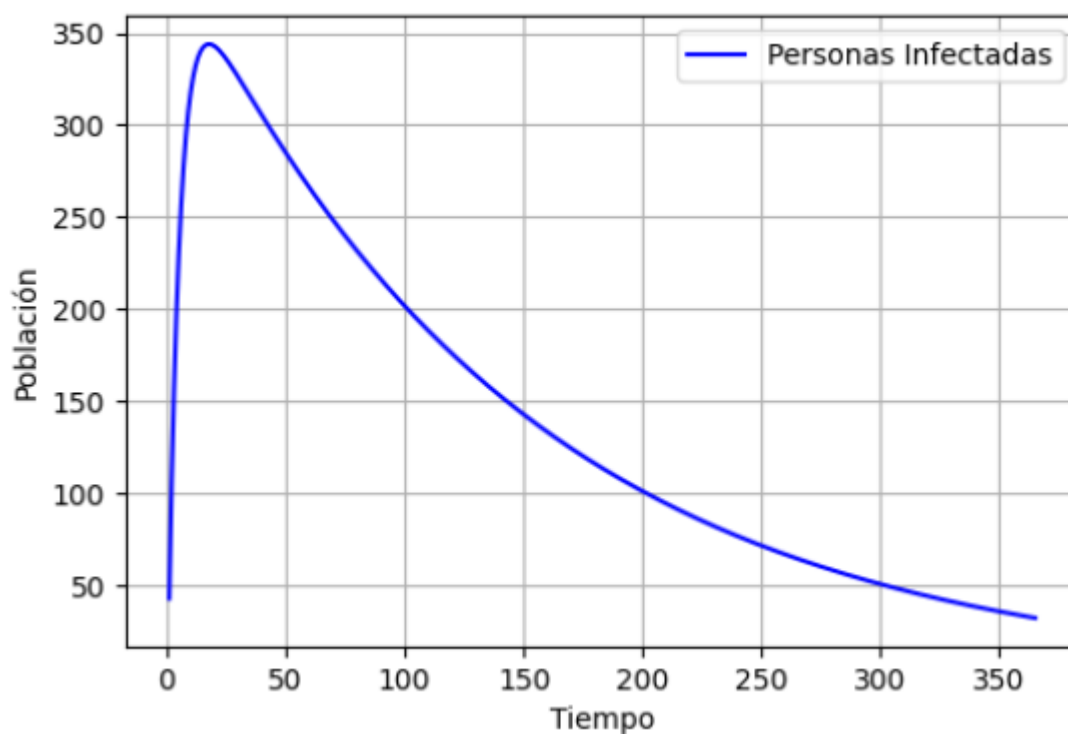
Para validar este modelo, se tomaron los datos de la semana epidemiológica 01 de panorama epidemiológico de dengue 2024 del gobierno de México.

ESTADO	PROBABLES		
	2023	2024	DNG 2023
TABASCO	32	150	0
COLIMA	21	70	0
GUANAJUATO	0	9	0
MORELOS	13	65	0
AGUASCALIENTES	0	1	0
BAJA CALIFORNIA	1	0	0
BAJA CALIFORNIA SUR	8	34	0
CAMPECHE	8	56	0
COAHUILA	0	0	0
CHIAPAS	15	45	0
CHIHUAHUA	0	0	0
CIUDAD DE MÉXICO	0	10	0
DURANGO	0	2	0
GUERRERO	21	277	0
HIDALGO	0	3	0
JALISCO	7	16	0
MÉXICO	2	10	0
MICHOACÁN	6	26	0
NAYARIT	2	4	0
NUEVO LEÓN	2	4	0
OAXACA	26	43	0
PUEBLA	2	12	0
QUERÉTARO	0	3	0
QUINTANA ROO	84	155	1
SAN LUIS POTOSÍ	1	7	0

Como se puede ver a Oaxaca con 43 casos probable como inicio en la primera semana, este valor de 43 también será tomado como punto inicial en la simulación.

ESTADO	PROBABLES		
	2023	2024	DNG 2023
GUERRERO	114	3,123	2
TABASCO	184	1,220	7
COLIMA	74	460	9
VERACRUZ	216	618	18
CAMPECHE	54	486	4
CHIAPAS	94	303	8
BAJA CALIFORNIA SUR	45	192	2
MORELOS	61	569	2
YUCATÁN	251	415	24
MICHOACÁN	36	188	4
QUINTANA ROO	524	1,160	63
SINALOA	51	90	4
NAYARIT	20	96	0
JALISCO	35	192	0
OAXACA	146	350	2
TAMAULIPAS	38	56	0
PUEBLA	22	86	2
SAN LUIS POTOSÍ	5	59	0

En la tercera semana se estima que haya alrededor de 350 casos probables.



En esta grafica se toma como punto inicial el número de casos que son 43, y se puede ver un rápido aumento hasta casi los 350 casos probables, que es un valor cercano al estimado.

INTERPRETACIÓN DE RESULTADOS

Para poder llevar a cabo las comparaciones de los años 2020, 2021, 2022, 2023 y 2024, utilizamos unos datos reales, los cuales, al momento de nosotros hacer la simulación, podemos determinar que tanto se aproximan las gráficas que generamos nosotros con el software con los datos reales.

Estos datos reales son:

PARA EL AÑO 2020:

'M': 15871556, población de personas

'N': 33967889, Población de mosquitos

'a': 0.32, tasa de picadura por mosquitos

'p': 0.27, probabilidad de infección por picadura en humanos

'c': 0.7, probabilidad de infección por picadura en mosquitos

'g': 0.16, tasa de recuperación de persona

'v': 0.06667

'X0': 80,

'V0': 1000

PARA EL AÑO 2021

'M': 16000000, población de personas

'N': 4000000, Población de mosquitos

'a': 0.37, tasa de picadura por mosquitos

'p': 0.21, probabilidad de infección por picadura en humanos

'c': 0.081, probabilidad de infección por picadura en mosquitos

'g': 0.144, tasa de recuperación de persona

'v': 0.068,

'P0': 100,

'V0': 1200

PARA EL AÑO 2022

'M': 16200000, población de personas

'N': 4100000, Población de mosquitos

'a': 0.30, tasa de picadura por mosquitos

'p': 0.25, probabilidad de infección por picadura en humanos

'c': 0.95, probabilidad de infección por picadura en mosquitos

'g': 0.145, tasa de recuperación de persona

'v': 0.0666

'P0': 90

'V0': 1100

PARA EL AÑO 2023

'M': 16400000, población de personas

'N': 4200000, Población de mosquitos

'a': 0.376, tasa de picadura por mosquitos

'p': 0.234, probabilidad de infección por picadura en humanos

'c': 0.082, probabilidad de infección por picadura en mosquitos

'g': 0.147, tasa de recuperación de persona

'v': 0.069

'P0': 100

'V0': 1300

PARA EL AÑO 2024

'M': 19000000, población de personas

'N': 4300000, Población de mosquitos

'a': 0.35, tasa de picadura por mosquitos

'p': 0.22688, probabilidad de infección por picadura en humanos

'c': 0.084, probabilidad de infección por picadura en mosquitos

'g': 0.146, tasa de recuperación de persona

'v': 0.069

'P0': 86

'V0': 1000

Pantalla de inicio

SOFTWARE DE SIMULACIÓN - VECTORES (Mosquitos) Y PERSONAS CONTAGIADAS POR DENGUE

¡BIENVENIDO!

ACERCA DE ESTE PROYECTO:

Pronostica la dinámica de transmisión del virus del dengue en el estado de Oaxaca a través del modelo matemático Ross-Macdonald.

Se le proporcionará resultados precisos sobre el número de personas y mosquitos infectados, así como resultados sobre cómo es la evolución de la epidemia en el estado.

MODELO ROSS-MACDONALD:

Este modelo consta de dos ecuaciones diferenciales y consiste en la interacción de 4 poblaciones distribuidas de la siguiente manera: la población de humanos y la población de mosquito y estas dos a su vez se dividen en susceptibles e infectados, esto formando cuatro grupos...

Además mediante el uso de ecuaciones diferenciales, aproxima el número de personas infectadas y mosquitos infectados en un tiempo determinado...

IR A LA SIMULACIÓN

Pantalla de inicio de simulación

Software de Simulación

Software de Simulación - Vectores (Mosquitos) y Personas Contagiados Por Dengue

Años disponibles ☐ 2020 ☐ 2021 ☐ 2022 ☐ 2023

Poblacion de Mosquitos:

Poblacion de Personas:

Tasa de Picadura por Mosquito:

Probabilidad de Infección por Picadura en Humanos:

Probabilidad de Infección Por Picadura en Mosquitos:

Tasa de Recuperacion de Personas:

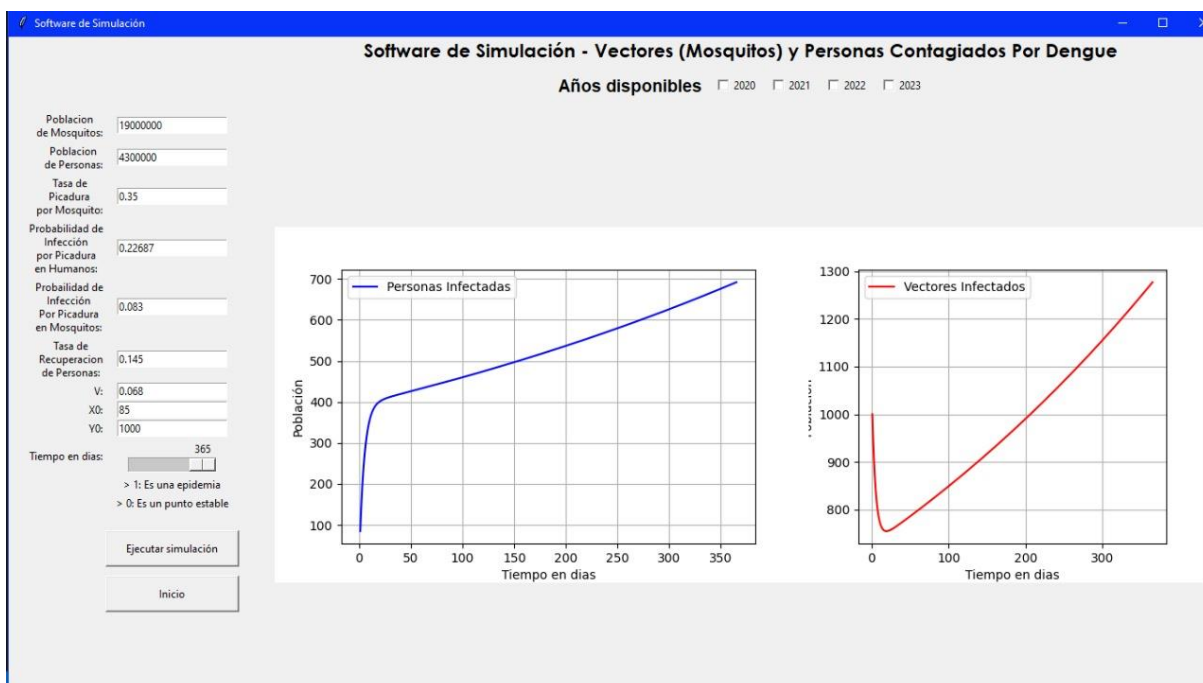
V:

X0:

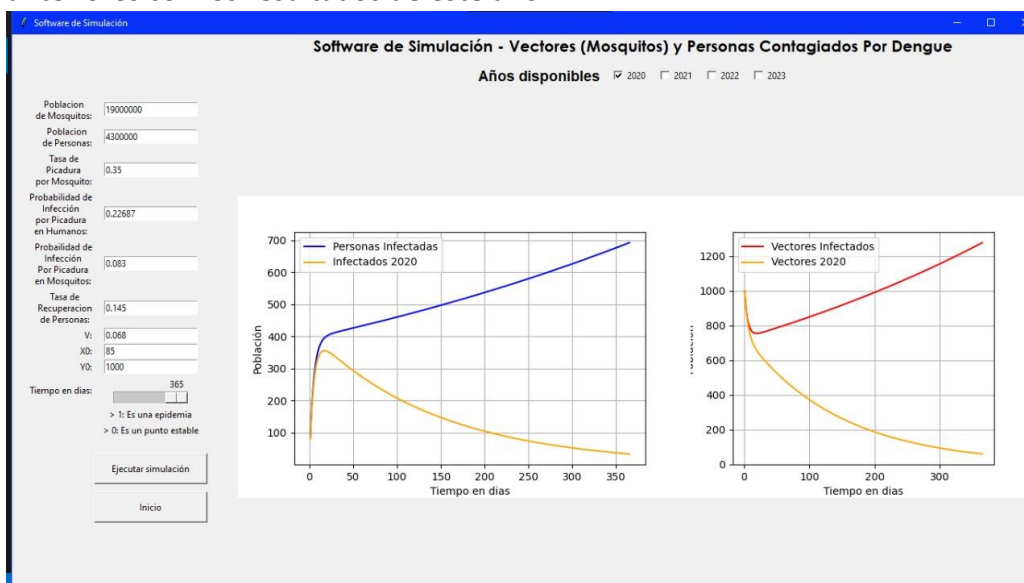
Y0:

Tiempo en dias:

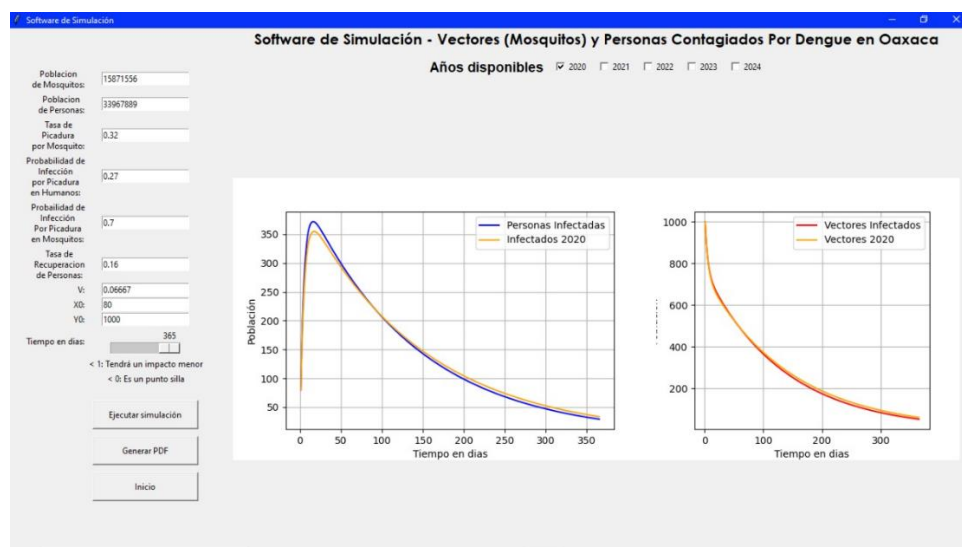
Cuando se ejecuta la simulación aparecen las 2 graficas, una corresponde a la gráfica de personas infectadas y la otra grafica corresponde a los vectores(mosquitos) infectados.



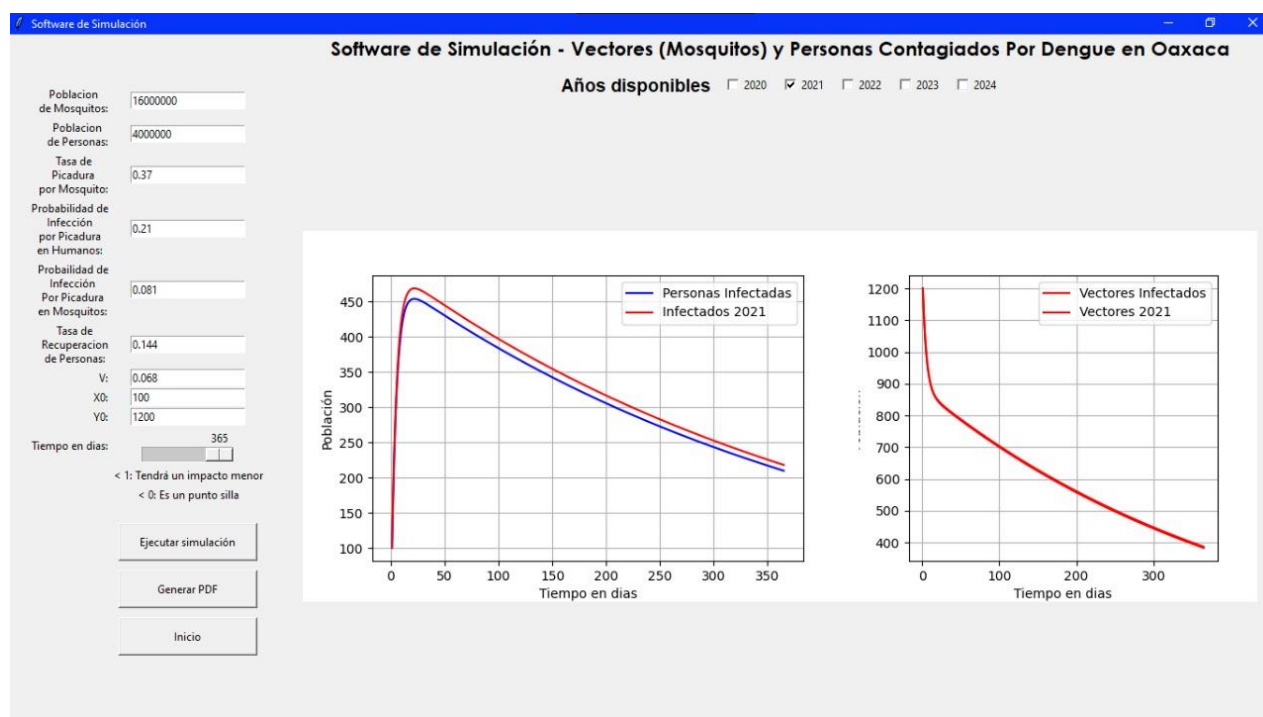
En la parte superior de la pantalla se puede visualizar una seccion de años disponibles, el usuario podrá seleccionar los años con los que podrá comparar los resultados de años anteriores con los resultados de este año.



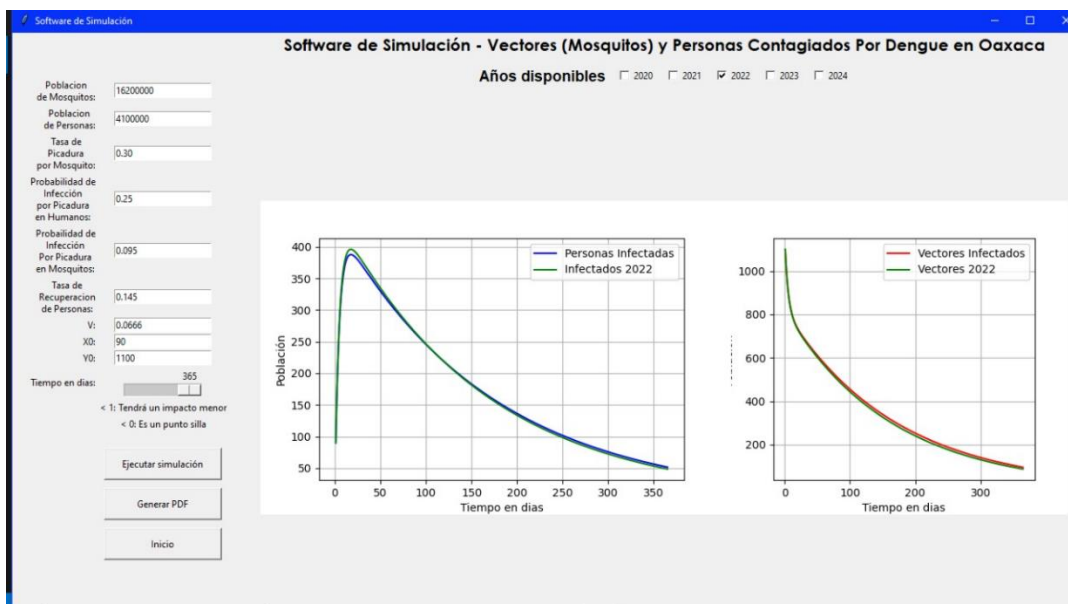
Si se desea comparar los datos de otros años con los datos que hay actualmente, solo se tiene que marcar el año que se desea comprar, en este caso primero se marcó la casilla con el año 2020, después de haber marcado la casilla, se va a pintar otra grafica en ambas graficas, esa nueva grafica corresponderá a la del año 2020, en este caso, y con eso se puede hacer la comparación.



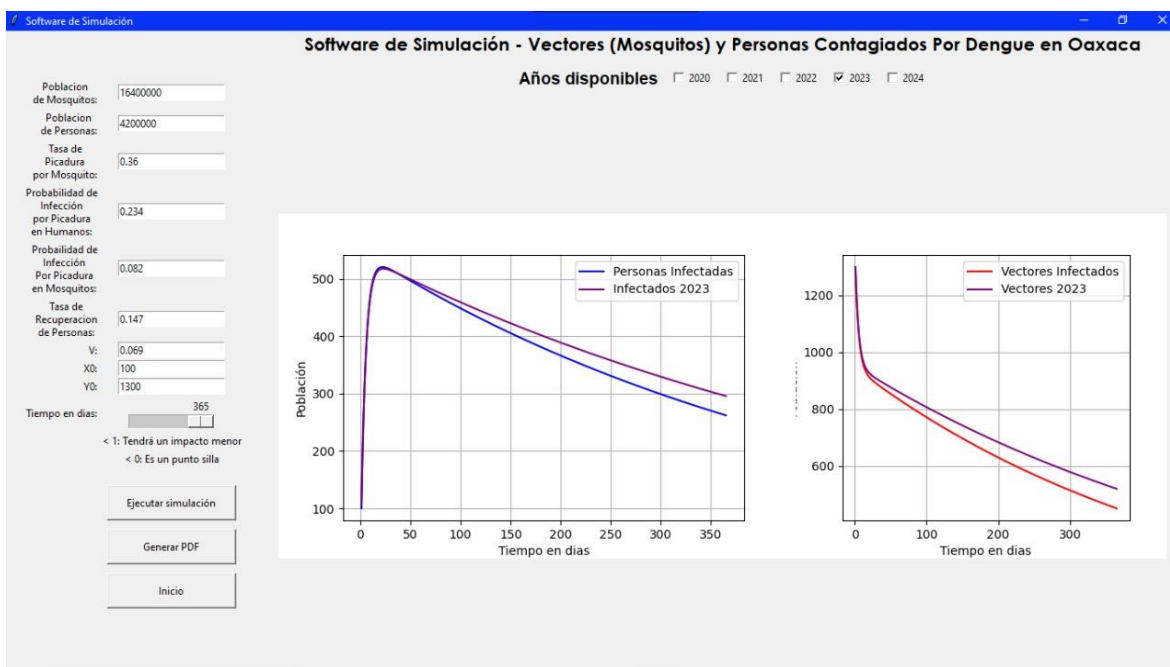
En este caso se marca la casilla con el año 2021, se pinta otra grafica y se pueden comparar los resultados.



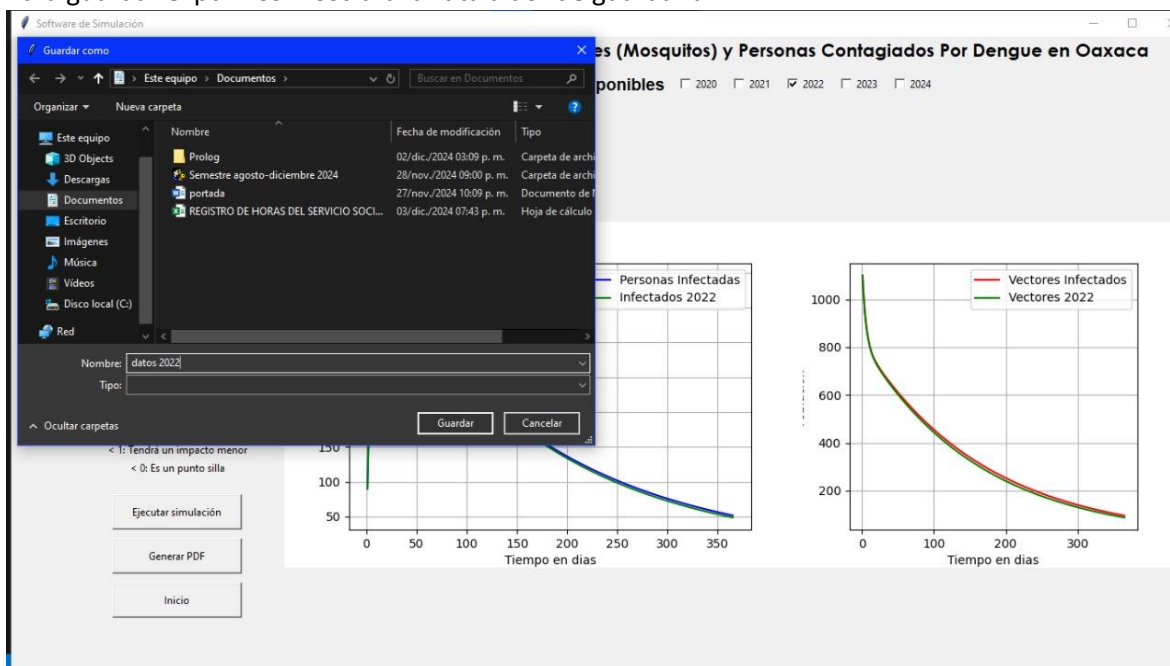
En este caso se marca la casilla con el año 2022, se pinta otra gráfica y se pueden comparar los resultados.



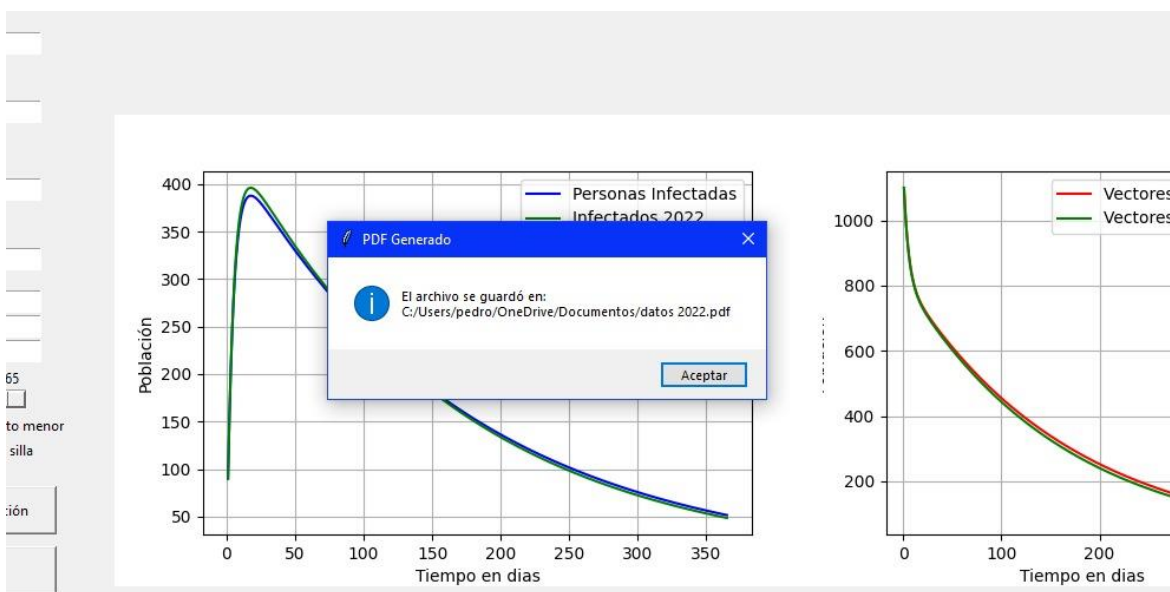
En este caso se marca la casilla con el año 2023, se pinta otra gráfica y se pueden comparar los resultados.



Para guardar el pdf nos mostrara la ruta a donde guardarla



Cuando finalice nos mostrara un mensaje donde nos dirá que se a realizado



CONCLUSIÓN

El modelo Ross-Macdonald ha demostrado ser una herramienta valiosa para entender la propagación de enfermedades transmitidas por vectores, incluyendo el virus del Dengue. El modelo proporciona un marco para comprender cómo la interacción entre los humanos y los vectores (en este caso, los mosquitos *Aedes aegypti*) puede conducir a brotes de la enfermedad. Al aplicar el modelo al Dengue, se puede estudiar varios factores que contribuyen a la propagación del virus, como la densidad de la población de mosquitos, las tasas de picadura, la duración del ciclo de vida del mosquito, entre otros. Esto permite evaluar el impacto potencial de diferentes estrategias de control, como la reducción de la población de mosquitos o la interrupción de su ciclo de vida.

Es importante señalar que, aunque el modelo Ross-Macdonald ofrece una visión valiosa, tiene ciertas limitaciones en el contexto del Dengue. Por ejemplo, el modelo asume que todos los individuos tienen la misma probabilidad de ser picados por mosquitos, lo que puede no ser cierto en la realidad, donde pueden existir variaciones significativas en la exposición al mosquito. Tampoco contempla la posibilidad de inmunidad adquirida, que podría influir en la dinámica de la enfermedad a largo plazo.

En resumen, el modelo Ross-Macdonald proporciona un marco sólido para entender la propagación del Dengue y evaluar las posibles estrategias de control. Sin embargo, a pesar de estas limitaciones, el modelo continúa siendo una herramienta útil.

REFERENCIAS

Mathematical model for the dynamic transmission of the Zika virus in Honduras. (s. f.). *Rev. Cient. Esc. Univ. Cienc. Salud.* 2018.

La historia del dengue aún no termina | Revista MVZ Córdoba. (s. f.).
<https://revistamvz.unicordoba.edu.co/article/view/1597/2295#content/>

Dengue (fiebre hemorrágica del dengue). (s. f.).
[https://www.health.ny.gov/es/diseases/communicable/dengue_fever/fact_sheet.htm#](https://www.health.ny.gov/es/diseases/communicable/dengue_fever/fact_sheet.htm#text=La%20forma%20hemorr%C3%A1gica%20del%20dengue,por%20ciento%20de%20los%20casos.)

text=La%20forma%20hemorr%C3%A1gica%20del%20dengue,por%20ciento%
20de%20los%20casos.

IMSS. (s. f.-b). <https://www.imss.gob.mx/salud-en-linea/dengue>

Dengue. (2024, 24 mayo). OPS/OMS | Organización Panamericana de la Salud.
<https://www.paho.org/es/temas/dengue>

De Salud, S. (s. f.). *Panorama epidemiológico de dengue 2024.* gob.mx.
<https://www.gob.mx/salud/documentos/panorama-epidemiologico-de-dengue-2024>