

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS MATEMÁTICAS
E.A.P. DE COMPUTACIÓN CIENTÍFICA

Modelo matemático de la dinámica de la malaria

TESINA

Para optar el Título Profesional de Licenciada en Computación
Científica

AUTOR

Flor de Mara Silvestre Manco

Lima - Perú

2016

Resumen

MODELO MATEMÁTICO DE LA DINÁMICA DE LA MALARIA

Silvestre Manco, Flor de María

Diciembre - 2016

La malaria es una enfermedad causada por parásitos del género Plasmodium, que se transmiten al ser humano por la picadura de mosquitos hembra infectados del género Anopheles. El presente trabajo, propone un modelo matemático que describe la dinámica de la malaria, formado por ecuaciones diferenciales ordinarias (ODEs). Los resultados muestran que si el número de reproducción R_0 es menor que 1, entonces el punto de equilibrio libre de enfermedad es estable, por lo tanto la enfermedad desaparece. Si R_0 es mayor que 1, entonces el punto de equilibrio libre de enfermedad es inestable, por lo tanto la enfermedad se propaga. Se ha realizado simulaciones numéricas con el software matemático Matlab. Estas simulaciones muestran el comportamiento de las poblaciones en el tiempo y la estabilidad de los puntos de equilibrio libre de enfermedad y endémicos.

Palabras Clave: Malaria
Sistema dinámico
Ecuación diferencial ordinaria
Puntos de equilibrio
Modelo endémico
Número de reproducción

Abstract

MATHEMATICAL MODEL OF DYNAMICS OF MALARIA

Silvestre Manco, Flor de María

Diciembre - 2016

Malaria is a disease caused by the Plasmodium parasites, that are transmitted to people through the bites of infected female Anopheles mosquitoes. The present work, proponed a mathematical model describes the dynamics of malaria, formado by ordinary differential equations(ODEs). Our results show that if the reproduction number, R_0 , is less than 1, the disease-free equilibrium point is stable, so that the disease dies out. If R_0 is larger than 1, then the disease-free equilibrium is unstable,so that the disease spread. Numerical simulations have been carried out with Matlab mathematical software. These simulations show the behavior of the populations in time and the stability of disease-free and endemic equilibrium.

Palabras Clave: Malaria
Dynamic system
Ordinary differential equation
Equilibrium Points
Endemic Model
Reproduction Number

Índice general

Resumen	2
Abstract	3
Introducción	5
1.1 Situación Problemática de la Malaria	6
1.2 Formulación del Problema	8
1.3 Objetivos	8
1.3.1 Objetivos generales	8
1.3.2 Objetivos específicos	8
Marco Teórico	9
2.1 Antecedentes de investigación	9
2.2 Bases teóricas	9
Modelo Matemático Epidemiológico de la Malaria	12
3.1 Parámetros del Modelo	12
3.2 Modelo SIRS con nacimientos y muerte natural	13
3.2.1 Existencia y Unicidad	14
3.2.2. Estado de Equilibrio del Modelo	16
Resultados y Discusión	21
4.1 Análisis, interpretación y discusión de resultados	21
Conclusiones	27
Apéndice	28
Bibliografía	35

Capítulo 1

Introducción

La malaria ó paludismo es una enfermedad parasitaria endémica causada por parásitos del género Plasmodium, que se transmiten al ser humano por la picadura de mosquitos hembra infectados del género Anopheles, los llamados vectores del paludismo.

Esta enfermedad sigue siendo una de las infecciones humanas más frecuentes y letales en todo el mundo. Según datos de la Organización Mundial de la Salud (OMS), se calcula que el 40% de la población mundial vive en zonas de malaria endémica. En el 2015 se registraron 214 millones de casos de la enfermedad, que costaron la vida a 438.000 personas, siendo el África Subsahariana el que soportó la carga más elevada con un 89% de casos de la enfermedad y un 91% de muertes por esta causa. Estos datos confirman la necesidad de seguir realizando investigaciones científicas desde los diferentes campos de la ciencia para reducir el impacto de la enfermedad y quizás con el tiempo su erradicación.

En el presente informe trataré el tema visto desde un enfoque matemático-epidemiológico, por ello no puedo dejar de mencionar a uno de los pioneros en utilizar un modelo matemático para predecir el comportamiento de la transmisión de la malaria, fue el británico nacido en la India, Ronald Ross, que es recordado principalmente por su descubrimiento del vector transmisor de la malaria. Ross planteó que era necesario tomar en cuenta una serie de variables y construir con ellas un modelo para predecir el comportamiento del brote infeccioso de malaria. En los años que le continuaron a este estudio pionero se dió un gran desarrollo de los modelos matemáticos que trataban la transmisión de las enfermedades en las poblaciones humanas. Uno de los trabajos mas trascendentes y que hasta hoy en día nos sirven de guía, son las ecuaciones conocidas como el modelo SIR, desarrollado por Kermack y McKendrick en 1927, este modelo implica una iteración entre huéspedes humanos infectados y vectores mosquitos infectados. En la actualidad los matemáticos epidemiólogos utilizan para sus investigaciones modelos como SIR con nacimientos y muertes, SIS, SIRS, SEIS, SEIR, MSIR, entre otros que en realidad son una evolución de los modelos pioneros.

En esta tesina presento el modelo SIRS (Susceptibles - Infectados - Recuperados - Susceptibles), como la representación de la dinámica de la enfermedad de la malaria y para su desarrollo seguiré la siguiente estructura: en el primer capítulo, se detalla

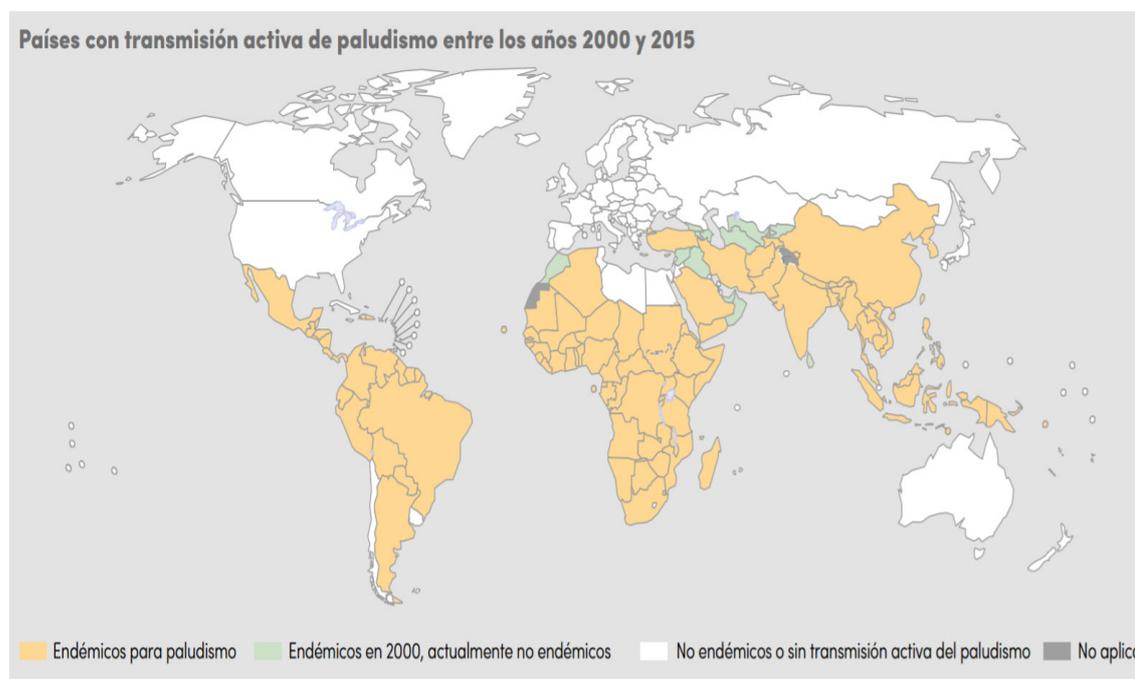
la situación problemática de la Malaria, tanto en el mundo como en el Perú; en el segundo capítulo, se realiza una breve introducción de la teoría matemática que sirve para el estudio y el análisis de un sistema dinámico y también se mencionan algunas definiciones, teoremas y lemas que se utilizaron a lo largo del trabajo.

En el tercer capítulo se desarrolla el análisis del modelo epidemiológico SIRS con nacimientos y muertes, basándose en la teoría expuesta en el capítulo anterior, donde verificaré la existencia y unicidad, determinaré los puntos críticos y el número básico de reproducción y la estabilidad e inestabilidad. Finalmente en el cuarto capítulo realizaré un simulación computacional con la ayuda del software Matlab y la interpretación de los resultados obtenidos mediante el método de Runge Kutta.

1.1 Situación Problemática de la Malaria

Situación Mundial

En mayo del 2015 la Asamblea Mundial de la Salud (OMS) adoptó una estrategia mundial contra la malaria. Esta tenía por objeto reducir en un 40 % la carga mundial de morbilidad para 2020, y en al menos un 90 % de aquí al 2030.[1]



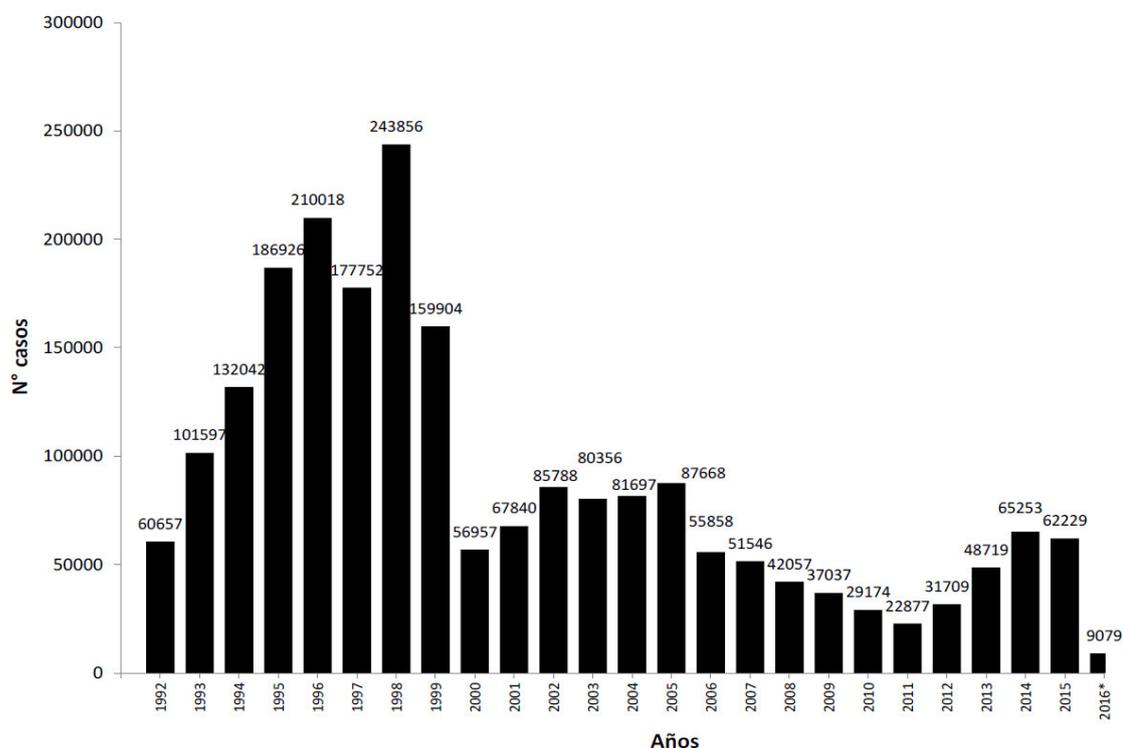
Entre los años 2000 y 2013, la tasa mundial de mortalidad estimada por malaria cayó un 47 %. La OMS calcula que en 2013 la malaria provocó la muerte de unas 584.000 personas. Actualmente existe transmisión endémica de malaria en 97 países y territorios. Se estima que el 82 % de los casos y el 90 % de las muertes tienen lugar en África. Las muertes en niños menores de 5 años equivalen, según las estimaciones de la OMS,

al 78 % del total. En 2013 el 80 % de los casos de malaria ocurrieron en 18 países, con la República Democrática del Congo y Nigeria aportando el 39 % de los casos a nivel mundial y más del 80 % de las muertes estimadas por malaria se concentran en tan solo 16 países.[1]

Situación en América y en el Perú

En América, el 38,4 % de la población vive en zonas de condiciones ecológicas propicias para la transmisión de la malaria. En el año 2000, de los 1,14 millones de casos notificados, el 53,6 % se presentaron en el Brasil, seguido por 9,45 % en Colombia, 8,65 % en el Ecuador, 6,12 % en el Perú, 4,68 % en Guatemala, 3,08 % en Honduras, 2,76 % en Bolivia, 2,61 % en Venezuela, 2,11 % en Guyana, 1,48 % en Haití y 1,15 % en Surinam.

La malaria es una enfermedad endémica en el Perú con un comportamiento reemergente desde el año 2012, especialmente, en la Amazonía, con una ligera reducción el año 2015. Durante el fenómeno de El Niño entre 1997-1998, el país alcanzó el máximo pico de malaria en su historia, más de 240 mil casos en el año 1998, desde entonces, se redujo sistemáticamente, con picos en el año 2002 y 2005, hasta el año 2011.[2]



1.2 Formulación del Problema

El modelo matemático para representar la dinámica de la malaria está basado en Ecuaciones Diferenciales Ordinarias (EDOs), que determinarán en que situación endémica se encuentra una determinada población en un tiempo determinado.

Existen varios modelos matemáticos que analizan la dinámica epidemiológica de diferentes enfermedades, incluida la malaria, pero existen algunos inconvenientes en la formulación de estos modelos matemáticos porque estos están orientados a la realidad de una determinada población que se encuentra en una situación endémica distinta de otras poblaciones, es por eso que se debe tener cuidado al momento de elegir un modelo específico.

La malaria ha sido desde hace muchos años objeto de estudio de los matemáticos epidemiológicos que la han analizado con diversos modelos matemáticos, como SIR, SIS, SIRE, SIRS, entre otros. El enfoque que tomaré en cuenta para el presente informe será el modelo matemático epidemiológico SIRS (Susceptibles - Infectados - Recuperados - Susceptibles), este modelo permite que un individuo después de infectado y recuperado vuelva nuevamente a ser susceptible, esto es, los individuos poseen inmunidad temporaria a la enfermedad, pudiendo ser reinfectados.

1.3 Objetivos

1.3.1 Objetivos generales

El presente trabajo tiene como objetivo encontrar soluciones para la prevención y el control de la enfermedad de la malaria mediante la aplicación de modelos matemáticos que conducen a un sistema de ecuaciones diferenciales ordinarias (ODEs).

1.3.2 Objetivos específicos

Los objetivos específicos es representar un modelo matemático epidemiológico óptimo, para ello debemos:

- Primero reconocer los parámetros con los que se efectuará el análisis ya que estos son importantes en la resolución del modelo.
- Realizar un análisis del sistema dinámico que me permita saber si los puntos de equilibrio son biológicamente factibles, es decir, si existen.
- Reconocer cuando un punto de equilibrio es libre de enfermedad ó endémico.
- Realizar un estudio del análisis de estabilidad local de los puntos de equilibrio para ello utilizaremos diferentes criterios.
- Formular un programa en Matlab con la finalidad de encontrar las soluciones del sistema dinámico que me permitan saber cuando una población susceptible se encuentra en situaciones de epidemia.

Capítulo 2

Marco Teórico

2.1 Antecedentes de investigación

Desde décadas atrás existen diversos antecedentes que demuestran los esfuerzos de los científicos que desde sus diferentes campos estudiaron la dinámica de la malaria con el objetivo de erradicar la enfermedad y evitar la propagación. Existen en la actualidad diversos artículos científicos que se enfocan en el estudio de la transmisión de la malaria mediante diversos modelos matemáticos. A continuación cito algunos de estos modelos:

- Modelling of Malaria Transmission in North Senatorial Zone of Taraba State Nigeria.[4]
- Analysis of age-structure malaria transmission model.[5]

2.2 Bases teóricas

Se realizará la formulación del modelo matemático de la transmisión de la malaria mediante un sistema de Ecuaciones Diferenciales Ordinarias, para ello se utilizará varios parámetros que son importantes en el desarrollo del sistema porque de estos dependen los resultados. A continuación se analizará la estabilidad del modelo que se resolverá mediante algunos teoremas, lemas y definiciones que se formulan en este capítulo.

LEMA 1: Sea $F : \mathbb{R}_+^n \longrightarrow \mathbb{R}_+^n$

$$F(x) = (F_1(x), F_2(x), \dots, F_n(x)), \quad x = (x_1, x_2, \dots, x_n)$$

son continuas y tienen derivadas parciales $\frac{\partial F_j}{\partial x_k}$ continuas en \mathbb{R}_+^n , $\forall j, k = 1, 2, \dots, n$. Entonces, F es localmente Lipschitz continua en \mathbb{R}_+^n . [6]

Teorema 1: Sea $F : \mathbb{R}_+^n \longrightarrow \mathbb{R}_+^n$ localmente Lipschitz continua para cada $j = \overline{1, n}$ y satisfice,

$$F_j \geq 0(x) \quad \text{para cualquier } x \in \mathbb{R}_+^n, x_j = 0$$

entonces para cada $x_0 \in \mathbb{R}_+^n$, existe una única solución de $x' = F(x)$ con $x(0) = x_0$ en \mathbb{R}_+^n donde está definida en algún intervalo $(0, b]$ con $b \in (0, \infty]$. Si $b < \infty$ entonces,

$$\sup_{0 \leq t \leq b} \sum_{j=1}^n x_j(t) = \infty \quad [6]$$

DEFINICIÓN DEL SISTEMA AUTÓNOMO

Un sistema autónomo plano es un sistema de dos ecuaciones diferenciales de la forma:

$$\begin{cases} \frac{dx}{dt} = F(x, y) \\ \frac{dy}{dt} = G(x, y) \end{cases} \quad (1)$$

donde supondremos que F y G son funciones continuas y con derivadas parciales de primer orden continuas en todo el plano. En este caso, las funciones F y G se dicen de clase C^1 en todo \mathbb{R}^2 ($F, G \in \mathbb{R}^2$). Estas condiciones sobre F y G garantizan la existencia y unicidad de la solución definida para todo $t \in \mathbb{R}$ del problema de valor inicial

$$\begin{cases} x' = F(x, y) & x(t_0) = x_0 \\ y' = G(x, y) & y(t_0) = y_0 \end{cases}$$

para cualquier $t_0 \in \mathbb{R}$ y $(x_0, y_0) \in \mathbb{R}^2$

ESTABILIDAD MEDIANTE LINEALIZACIÓN

Consideremos el sistema autónomo (1), con un punto crítico en (x_0, y_0) , tal que las funciones $F(x, y)$ y $G(x, y)$ sean de clase $C^1(\mathbb{R}^2)$. Entonces, aproximando $z = F(x, y)$ y $w = G(x, y)$ cerca del punto (x_0, y_0) por sus respectivos planos tangentes en dicho punto,

$$\begin{aligned} F(x, y) &\approx \frac{\partial F}{\partial x}(x_0, y_0)(x - x_0) + \frac{\partial F}{\partial y}(x_0, y_0)(y - y_0) \\ G(x, y) &\approx \frac{\partial G}{\partial x}(x_0, y_0)(x - x_0) + \frac{\partial G}{\partial y}(x_0, y_0)(y - y_0) \end{aligned}$$

podemos escribirlo como,

$$\begin{pmatrix} F(x, y) \\ G(x, y) \end{pmatrix} \approx A \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \quad \text{si } (x, y) \simeq (x_0, y_0),$$

donde A es la matriz jacobiana del campo $(F(x, y), G(x, y))^t$ en el punto (x_0, y_0) , es decir,

$$A = \begin{pmatrix} \frac{\partial F}{\partial x}(x_0, y_0) & \frac{\partial F}{\partial y}(x_0, y_0) \\ \frac{\partial G}{\partial x}(x_0, y_0) & \frac{\partial G}{\partial y}(x_0, y_0) \end{pmatrix}$$

De esta manera, podemos pensar que el sistema (1) se encuentra próximo al sistema lineal,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = A \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \quad (2)$$

cuando (x, y) está cerca de (x_0, y_0) , y por consiguiente, es natural esperar que el comportamiento de las trayectorias del sistema (1) cerca del punto crítico (x_0, y_0) sea similar al de las trayectorias del sistema linealizado (2).

El proceso descrito con anterioridad se denomina linealización y nótese que el cambio de variable sobre el sistema (2) transforma éste en el sistema lineal,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} \quad (3)$$

que tiene al punto $(0, 0)$ como punto de equilibrio.

En general, el punto de equilibrio (x_0, y_0) del sistema autónomo (1) hereda la estabilidad, y en algunos casos la naturaleza del punto de equilibrio (x_0, y_0) para el sistema lineal (2), es decir, la estabilidad del $(0, 0)$ para el sistema lineal (3).

PROPOSICIÓN 1 : Si el punto crítico $(0, 0)$ del sistema lineal (3) es aislado (es decir, si $\det(A) \neq 0$), entonces el punto crítico (x_0, y_0) del sistema (1) es aislado.

TEOREMA 2 : Teorema de Linealización de Liapunov y Poincaré

- a. El punto crítico (x_0, y_0) del sistema (1) es asintóticamente estable si y sólo si todos los autovalores de la matriz A poseen parte real negativa (esto es, si el punto crítico $(0, 0)$ del sistema (3) es asintóticamente estable).
- b. El punto crítico (x_0, y_0) del sistema (1) es inestable si y sólo si la matriz A del sistema posee un autovalor con parte real positiva (es decir, el punto crítico $(0, 0)$ es inestable para el sistema (3)).

Más aún, si los autovalores de A son distintos entre sí y distintos de cero se puede decir lo siguiente:

1. $\lambda_1 < \lambda_2 < 0$, entonces (x_0, y_0) es un nodo asintóticamente estable.
2. $\lambda_1 > \lambda_2 > 0$, entonces (x_0, y_0) es un nodo inestable.
3. $\lambda_1 < 0 < \lambda_2$, entonces (x_0, y_0) es un punto de silla.
4. λ_1 no es real y $Re(\lambda_1) < 0$ entonces (x_0, y_0) es un foco asintóticamente estable.
5. λ_1 no es real y $Re(\lambda_1) > 0$ entonces (x_0, y_0) es un foco inestable.

Capítulo 3

Modelo Matemático Epidemiológico de la Malaria

El modelo matemático que consideré para el estudio dinámico de la transmisión de la malaria, es el modelo SIRS, para su formulación se tendrá en cuenta que la población se dividirá en susceptibles (individuos que pueden ser infectados), infectados (individuos que son infectados por la enfermedad) y recuperados (individuos que se recuperan de la enfermedad). Además los individuos recuperados vuelven nuevamente a ser susceptibles, es decir, los individuos poseen inmunidad temporaria a la enfermedad.

3.1 Parámetros del Modelo

Vamos a mencionar los parámetros poblacionales, las tasas y las dinámicas que se utilizará para la formulación del modelo.

Poblaciones: Los valores que toman las poblaciones siempre serán mayores e iguales a cero, por tratarse de una población de individuos.

N : Población total.

S : Población de susceptibles.

I : Población de infectados.

R : Población de recuperados.

donde: $N = S + I + R$

Tasas: Los valores de las tasas serán únicamente entre cero y uno abierto, por tratarse de valores estadísticos.

Λ : Tasa per cápita de reclutamiento de la población susceptible.

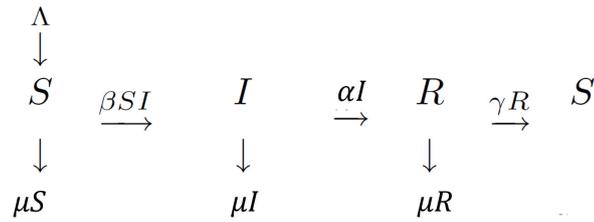
μ : Tasa per cápita de muerte natural.

β : Tasa de contacto.

α : Tasa per cápita de recuperación.

γ : Tasa per cápita de inmunidad pérdida.

Dinámicas: En el estudio de este modelo se observa un movimiento de individuos que ingresan ó salen de los estados de susceptible, infectado y recuperado. Este movimiento ó flujo son las dinámicas que se producen por la interacción de los parámetros poblaciones y tasas. A continuación un diagrama de transferencia para el modelo SIRS que representará el uso de estas dinámicas:



$-\beta SI$: Fracción de individuos que pasan del estado susceptible al infectado.

$-\alpha I$: Fracción de individuos que pasan del estado infectado al recuperado.

$-\gamma R$: Fracción de individuos que pasan del estado recuperado al susceptible

$-\mu S$: Fracción de individuos susceptibles que fallecen por muerte natural.

$-\mu I$: Fracción de individuos infectados que fallecen por muerte natural.

$-\mu R$: Fracción de individuos recuperados que fallecen por muerte natural.

3.2 Modelo SIRS con nacimientos y muerte natural

La formulación del modelo SIRS de acuerdo con el diagrama de transferencia, conduce al siguiente sistema de Ecuaciones Diferencias Ordinarias:

$$\left\{ \begin{array}{l} \frac{dS}{dt} = \Lambda - \beta SI - \mu S + \gamma R \\ \frac{dI}{dt} = \beta SI - (\mu + \alpha)I \\ \frac{dR}{dt} = \alpha I - (\mu + \gamma)R \end{array} \right. \quad (4)$$

Vamos analizar el sistema (4) cuando la población es constante, para esto hay que considerar $\Lambda = \mu$. Entonces el nuevo sistema de E.D.O. será de la siguiente forma:

$$\left\{ \begin{array}{l} \frac{dS}{dt} = \mu N - \beta SI - \mu S + \gamma R \\ \frac{dI}{dt} = \beta SI - (\mu + \alpha)I \\ \frac{dR}{dt} = \alpha I - (\mu + \gamma)R \end{array} \right. \quad (5)$$

con condiciones iniciales: $S(0) \geq 0$, $I(0) \geq 0$, $R(0) \geq 0$

3.2.1 Existencia y Unicidad

Antes de poder hablar de soluciones, hay que demostrar la existencia y unicidad del sistema (4) de E.D.O. para ello utilizaré el Lema 1 y Teorema 1 definidos en el capítulo 2.

Del LEMA 1:

Sea $F : \mathbb{R}_+^3 \rightarrow \mathbb{R}_+^3$

$$F(X) = (F_1(X), F_2(X), F_3(X)), \quad X = (S^*, I^*, R^*)$$

donde S, I y R son funciones continuas positivas y además $F_1(X), F_2(X), F_3(X)$ son funciones continuas.

Luego vamos a analizar las derivadas parciales de cada función.

- Sea $F_1(S, I, R) = \Lambda - \beta SI - \mu S + \gamma R$

$$\frac{\partial F_1}{\partial S} = -\beta I - \mu, \quad \frac{\partial F_1}{\partial I} = -\beta S, \quad \frac{\partial F_1}{\partial R} = \gamma$$

entonces $\frac{\partial F_1}{\partial S}, \frac{\partial F_1}{\partial I}, \frac{\partial F_1}{\partial R}$ son continuas.

- Sea $F_2(S, I, R) = \beta SI - (\mu + \alpha)I$

$$\frac{\partial F_2}{\partial S} = \beta I, \quad \frac{\partial F_2}{\partial I} = \beta S - (\mu + \alpha), \quad \frac{\partial F_2}{\partial R} = 0$$

entonces $\frac{\partial F_2}{\partial S}, \frac{\partial F_2}{\partial I}, \frac{\partial F_2}{\partial R}$ son continuas.

- Sea $F_3(S, I, R) = \alpha I - (\mu + \gamma)R$

$$\frac{\partial F_3}{\partial S} = 0, \quad \frac{\partial F_3}{\partial I} = \alpha, \quad \frac{\partial F_3}{\partial R} = -(\mu + \gamma)$$

entonces $\frac{\partial F_3}{\partial S}, \frac{\partial F_3}{\partial I}, \frac{\partial F_3}{\partial R}$ son continuas.

$\therefore F$ es localmente Lipschitz continua en \mathbb{R}_+^3 .

Del TEOREMA 1:

Sea $F : \mathbb{R}_+^3 \rightarrow \mathbb{R}^3$ localmente Lipschitz continua para cada $j = \overline{1, 3}$ y satisfice,

$$F_j(X) \geq 0 \quad \text{para cualquier } X \in \mathbb{R}_+^3, X_j = 0$$

entonces para cada $X_0 \in \mathbb{R}_+^3$, existe una única solución de $X' = F(X)$ con $X(0) = X_0$ en \mathbb{R}_+^3 donde está definida en algún intervalo $(0, b]$ con $b \in (0, \infty]$.

Hay que verificar que satisfice las condiciones del teorema 1.

- Sea $F_1(S, I, R) = \Lambda - \beta SI - \mu S + \gamma R$, donde $S = 0, I \geq 0, R \geq 0$
 $\rightarrow F_1(S, I, R) = \Lambda + \gamma R \geq 0$

- Sea $F_2(S, I, R) = \beta SI - (\mu + \alpha)I$ donde $S \geq 0, I = 0, R \geq 0$
 $\rightarrow F_2(S, I, R) = 0 \geq 0$

- Sea $F_3(S, I, R) = \alpha I - (\mu + \gamma)R$ donde $S \geq 0, I \geq 0, R = 0$
 $\rightarrow F_3(S, I, R) = \alpha I \geq 0$

$$\therefore F_j(X) \geq 0 \quad \forall j = 1, 2, 3$$

Si tenemos que :

$$b < \infty \rightarrow \sup_{0 \leq t \leq b} (S + I + R) = \infty$$

Supongamos, $b < \infty$ y sea $N=S+I+R$

$$N' = \Lambda - \mu S - \mu I - \mu R$$

$$N' = \Lambda - \mu(S + I + R)$$

$$N' = \Lambda - \mu N$$

Integrando la expresión:

$$\int_0^t N' dt = \int_0^t \Lambda dt - \int_0^t \mu N(s) ds$$

$$N(t) - N(0) = \Lambda t - \int_0^t \mu N(s) ds$$

$$N(t) = \Lambda t + N(0) - \int_0^t \mu N(s) ds$$

Por el Teorema de Gronwall:

$$N(t) \leq (\Lambda t + N(0))e^{-\mu t} = \frac{\Lambda t}{e^{\mu t}} + \frac{N(0)}{e^{\mu t}}$$

ya que, se cumple que:

a) $e^{\mu t} \geq 1 \rightarrow \frac{N(0)}{e^{\mu t}} \leq N(0)$

b) $\frac{\Lambda t}{e^{\mu t}} \rightarrow 0$ cuando $t \rightarrow \infty$

Entonces, se deduce que:

$$N(t) \leq (\Lambda t + N(0))e^{-\mu t}, \quad \forall t \in [0, b >$$

Sea $h(t) = \Lambda t + N(0) \leq M, \quad \forall t \in [0, b >$

ya que, $b < +\infty \rightarrow h(t) \leq M, \quad \forall t \in [0, b >$

entonces: $N(t) \leq M$ para algún $M > 0$

$$\sup\{S + I + R\} \leq M, \quad \forall t \in [0, b >$$

de aquí,

$$\sup\{S + I + R\} < \infty \quad (\rightarrow \leftarrow)$$

$$\therefore b = \infty$$

De esto se concluye que, el sistema (4) tiene una única solución definida en $[0, \infty > .\square$

3.2.2 Estado de Equilibrio del Modelo

Vamos a realizar el análisis cualitativo del sistema para garantizar la estabilidad del modelo matemático.

Puntos Críticos:

Sea el sistema (5) de E.D.O.

$$\begin{cases} \mu N - \beta SI - \mu S + \gamma R = 0 \quad \dots (1) \\ \beta SI - (\mu + \alpha)I = 0 \quad \dots (2) \\ \alpha I - (\mu + \gamma)R = 0 \quad \dots (3) \end{cases}$$

De la ecuación (2):

$$I(\beta S - (\mu + \alpha)) = 0$$

- Para: $I = 0$

$$\rightarrow S \neq \frac{\mu + \alpha}{\beta}$$

reemplazando en (1):

$$\mu N - \mu S + \gamma R = 0 \quad \dots (\star)$$

reemplazando en (3):

$$(\mu + \gamma)R = 0$$

$$\rightarrow R = 0 \dots (\star\star)$$

reemplazando $(\star\star)$ en (\star) :

$$\mu N - \mu S = 0$$

$$N - S = 0$$

$$\rightarrow S = N(0)$$

Por lo tanto, el sistema (5) tiene un punto de equilibrio libre de enfermedad $E_1^* = (S^0, I^0, R^0) = (N(0), 0, 0)$.

■ Para: $I \neq 0$

$$\rightarrow \beta S - (\mu + \alpha) = 0$$

$$S = \frac{\mu + \alpha}{\beta} \dots (\diamond)$$

despejando R de la ecuación (3):

$$R = \frac{\alpha I}{\mu + \gamma} \dots (\diamond\diamond)$$

reemplazando (\diamond) y $(\diamond\diamond)$ en la ecuación (1):

$$\mu N - \beta \left(\frac{\mu + \alpha}{\beta} \right) I - \mu \left(\frac{\mu + \alpha}{\beta} \right) + \gamma \left(\frac{\alpha I}{\mu + \gamma} \right) = 0$$

despejando I :

$$I = \frac{(\mu + \gamma) \left(N - \left(\frac{\mu + \alpha}{\beta} \right) \right)}{\mu + \gamma + \alpha}$$

así,

$$R = \frac{\alpha \left(N - \left(\frac{\mu + \alpha}{\beta} \right) \right)}{\mu + \gamma + \alpha}$$

Por lo tanto el sistema (5) tiene un punto de equilibrio endémico

$$E_2^* = (S^*, I^*, R^*) = \left(\frac{\mu + \alpha}{\beta}, \frac{(\mu + \gamma) \left(N(0) - \left(\frac{\mu + \alpha}{\beta} \right) \right)}{\mu + \gamma + \alpha}, \frac{\alpha \left(N(0) - \left(\frac{\mu + \alpha}{\beta} \right) \right)}{\mu + \gamma + \alpha} \right)$$

Para que el punto de equilibrio endémico exista debe cumplirse que $I^* > 0$:

$$\frac{(\mu + \gamma) \left(N(0) - \left(\frac{\mu + \alpha}{\beta} \right) \right)}{\mu + \gamma + \alpha} > 0$$

$$\rightarrow N(0) > \frac{\mu + \alpha}{\beta}$$

$$\rightarrow \underbrace{\frac{\beta N(0)}{\mu + \alpha}}_{R_0} > 1$$

Esto garantiza que el punto de equilibrio endémico existe, donde R_0 es el número básico de reproducción.

Linealización

Si asumimos que la población total N es constante, entonces:

$$N(t) \equiv N(0) = S(t) + I(t) + R(t)$$

de la ecuación, se obtiene que:

$$R(t) = N(0) - (S(t) + I(t))$$

si reemplazo este resultado en el sistema (5), podemos considerar el siguiente sistema bidimensional:

$$\begin{cases} \frac{dS}{dt} = (\mu + \gamma)N - \beta SI - (\mu + \gamma)S - \gamma I \\ \frac{dI}{dt} = \beta SI - (\mu + \alpha)I \end{cases} \quad (6)$$

Vamos a analizar la estabilidad de los puntos críticos, para ello se considera el sistema linealizado, dado por:

$$X' = J(\bar{S}, \bar{I})X$$

La matriz asociada será:

$$J(\bar{S}, \bar{I}) = \begin{bmatrix} -(\beta\bar{I} + \mu + \gamma) & -(\beta\bar{S} + \gamma) \\ \beta\bar{I} & \beta\bar{S} - (\mu + \alpha) \end{bmatrix}$$

donde (\bar{S}, \bar{I}) es un punto de equilibrio del sistema (6).

a) La matriz asociada a la linealización de E_1 es:

$$J(N(0), 0) = \begin{bmatrix} -(\mu + \gamma) & -(\beta N(0) + \gamma) \\ 0 & \beta N(0) - (\mu + \alpha) \end{bmatrix}$$

donde,

$$\det(J - \lambda I) = 0$$

$$\lambda^2 - \lambda(\beta N(0) - (2\mu + \alpha + \gamma)) - (\mu + \gamma)(\beta N(0) - (\alpha + \mu)) = 0$$

los autovalores son:

$$\lambda_1 = -(\mu + \gamma) < 0$$

$$\lambda_2 = \beta N(0) - (\alpha + \mu)$$

- Caso 1:

$$\lambda_2 = \beta N(0) - (\alpha + \mu) > 0$$

$$\underbrace{\frac{\beta N(0)}{\alpha + \mu}}_{R_0} > 1$$

Si $\lambda_1 < 0$ y $\lambda_2 > 0$ entonces $E_1 = (N(0), 0)$ es un punto inestable y no existe epidemia.

- Caso 2:

$$\lambda_2 = \beta N(0) - (\alpha + \mu) < 0$$

$$\underbrace{\frac{\beta N(0)}{\alpha + \mu}}_{R_0} < 1$$

Si $\lambda_1 < 0$ y $\lambda_2 < 0$ entonces $E_1 = (N(0), 0)$ es punto estable y no existe epidemia.

Por lo tanto, para que un punto de equilibrio libre de enfermedad $(N(0), 0, 0)$ sea estable, debe cumplirse que $N(0) < \frac{\alpha + \mu}{\beta}$.

b) La matriz asociada a la linealización de E_2 es:

$$J(S^*, I^*) = \begin{bmatrix} -(\beta I^* + \mu + \gamma) & -(\mu + \alpha + \gamma) \\ \beta I^* & 0 \end{bmatrix}$$

tenemos que,

$$\text{traza}J(S^*, I^*) = -(\beta I^* + \mu + \gamma)$$

$$\det J(S^*, I^*) = (\mu + \alpha + \gamma)\beta I^*$$

Por condiciones de estabilidad local del punto (S^*, I^*) será estable si, y solamente si,

i)

$$TJ(S^*, I^*) < 0$$

$$-(\beta I^* + \mu + \gamma) < 0$$

$$I^* > -\frac{\mu + \gamma}{\beta}$$

$$\frac{(\mu + \gamma)(N - S^*)}{\mu + \alpha + \gamma} > -\frac{\mu + \gamma}{\beta}$$

$$N - S^* > -\frac{\mu + \alpha + \gamma}{\beta}$$

$$N(0) > \frac{\gamma}{\beta}$$

ii)

$$\begin{aligned}DJ(S^*, I^*) &> 0 \\(\mu + \alpha + \gamma)\beta I^* &> 0 \\I^* &> 0 \\N &> S^* \\\frac{\beta N(0)}{\underbrace{\mu + \alpha}_{R_0}} &> 1\end{aligned}$$

Por lo tanto, para que un punto de equilibrio endémico (S^*, I^*, R^*) sea estable, debe cumplirse que $N(0) > \frac{\alpha + \mu}{\beta}$.

Capítulo 4

Resultados y Discusión

4.1 Análisis, interpretación y discusión de resultados

Vamos a complementar el análisis matemático con una simulación computacional mediante el entorno Matlab que nos permita analizar el comportamiento del modelo matemático, el cual está formado de ecuaciones diferenciales ordinarias, por ello se utilizó los métodos numéricos para aproximar la solución del sistema. Primero se realizará un análisis de los puntos de equilibrio con los mismos parámetros que se usarán para la representación gráfica del modelo.

Diagrama de Fases

Para realizar una representación gráfica del diagrama de fases, se utilizó las herramientas interactivas de MATLAB, como son: DFIELD8 para el estudio de ecuaciones diferenciales y PPLANE8 que traza campos vectoriales para sistemas autónomos.

De la primera ecuación del sistema (5), obtenemos la isoclina de S:

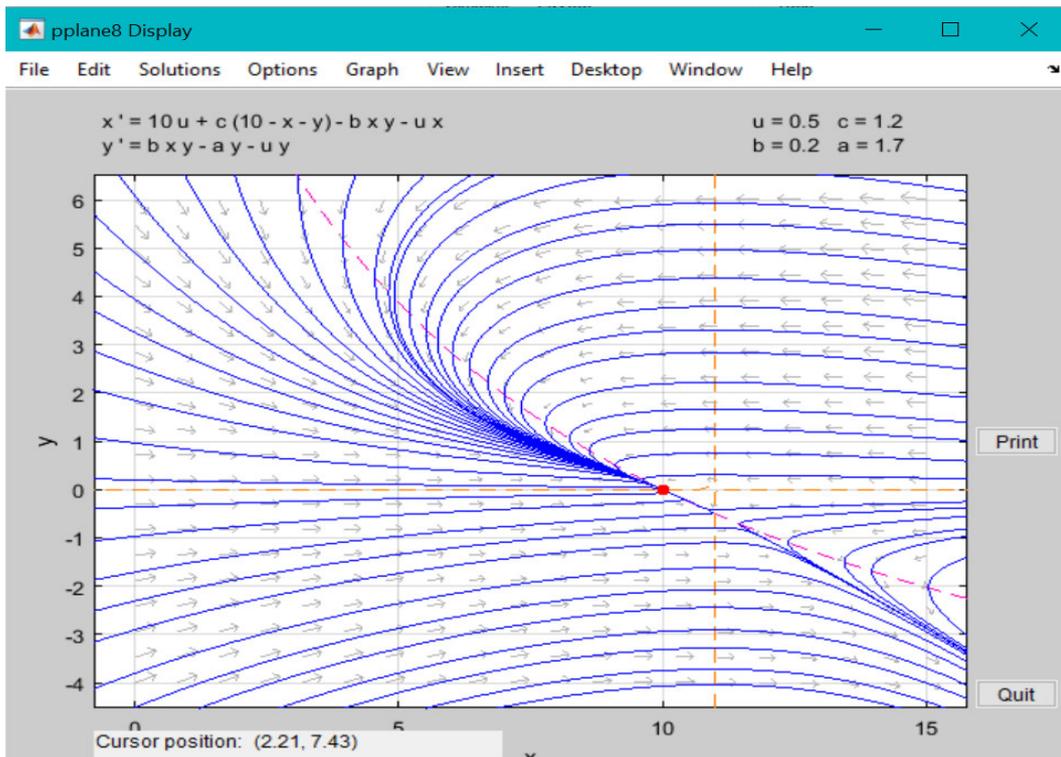
$$I = \frac{(\mu + \gamma)(N(0) - S)}{\gamma + \beta S}$$

De la segunda ecuación del sistema (5), obtenemos las isoclinas de I:

$$I = 0 \vee S = \frac{\mu + \alpha}{\beta}$$

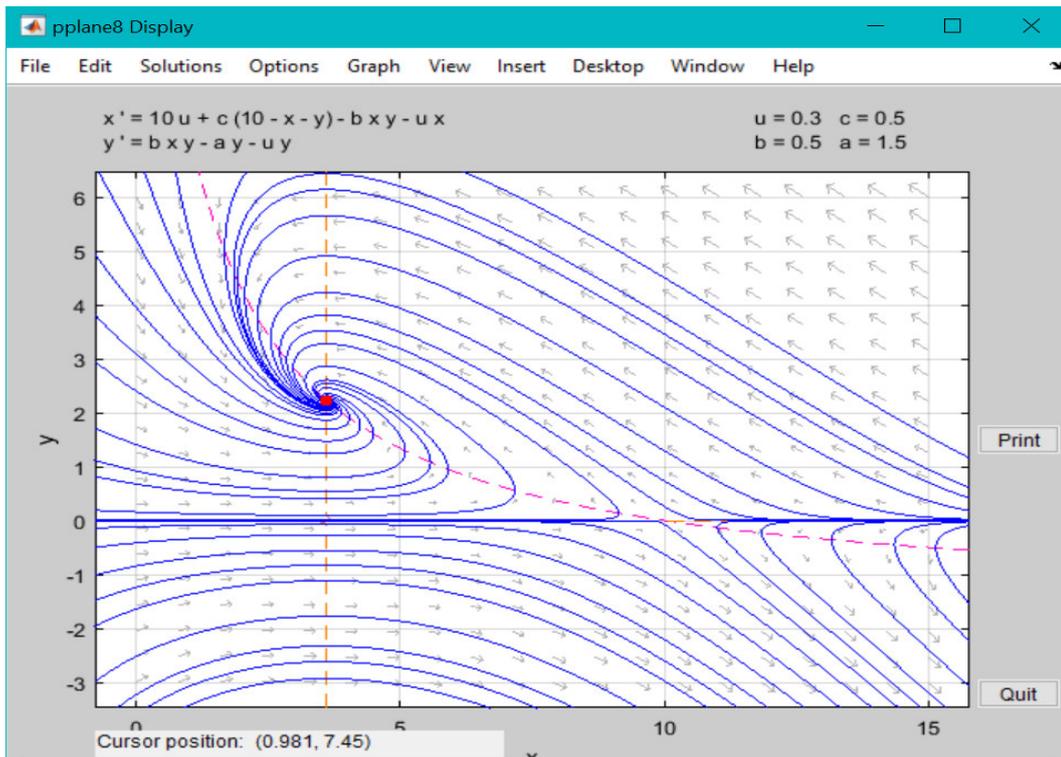
a) Para el primer caso tomaremos en cuenta los siguientes parámetros:

$$\beta = 0.2, \quad \mu = 0.5, \quad \alpha = 1.7, \quad \gamma = 1.2$$



La figura representa el diagrama de fases del sistema SIRS y sus respectivas isoclinas. Tenemos que $N < \frac{\alpha + \mu}{\beta}$ por lo tanto el punto de equilibrio libre de enfermedad (punto rojo) es biológicamente factible (existe) y además es un nodo asintóticamente estable.

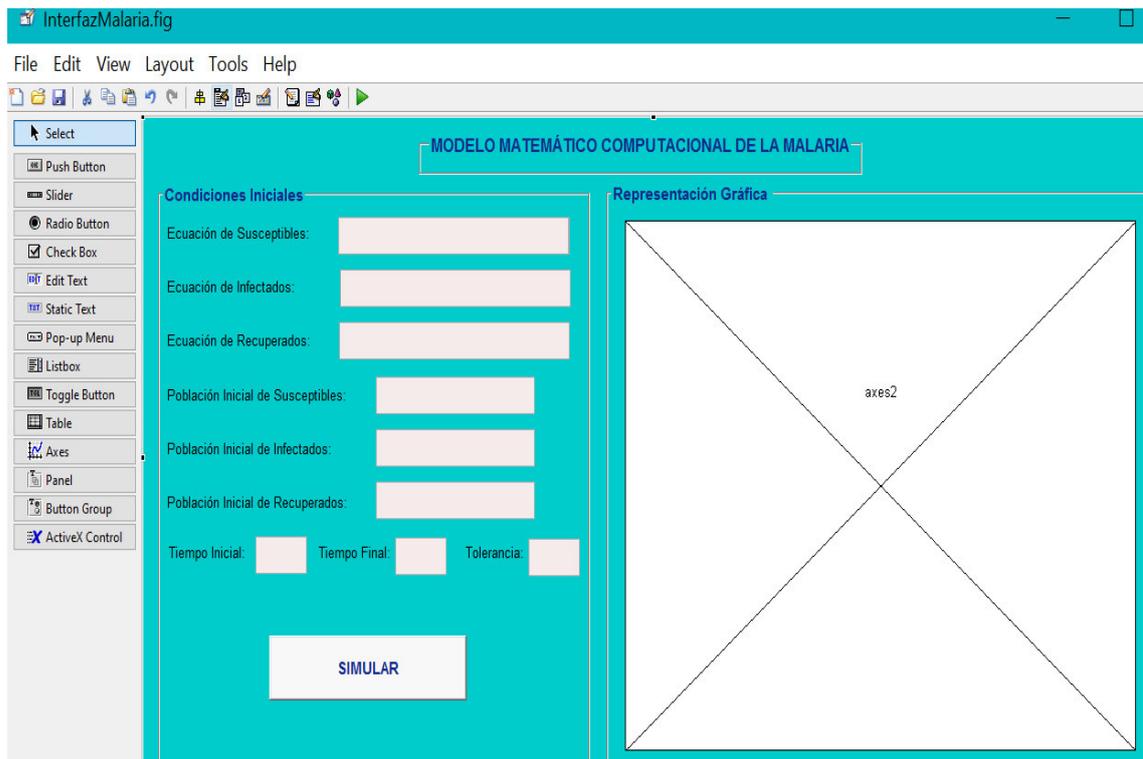
- b) Para el segundo caso tomaremos en cuenta los siguientes parámetros:
 $\beta = 0.5$, $\mu = 0.3$, $\alpha = 1.5$, $\gamma = 0.5$



La figura representa el diagrama de fases del sistema SIRS y sus respectivas isoclinas. Tenemos que $N > \frac{\alpha + \mu}{\beta}$ por lo tanto el punto de equilibrio endémico (punto rojo) es biológicamente factible (existe) y además es una espiral estable.

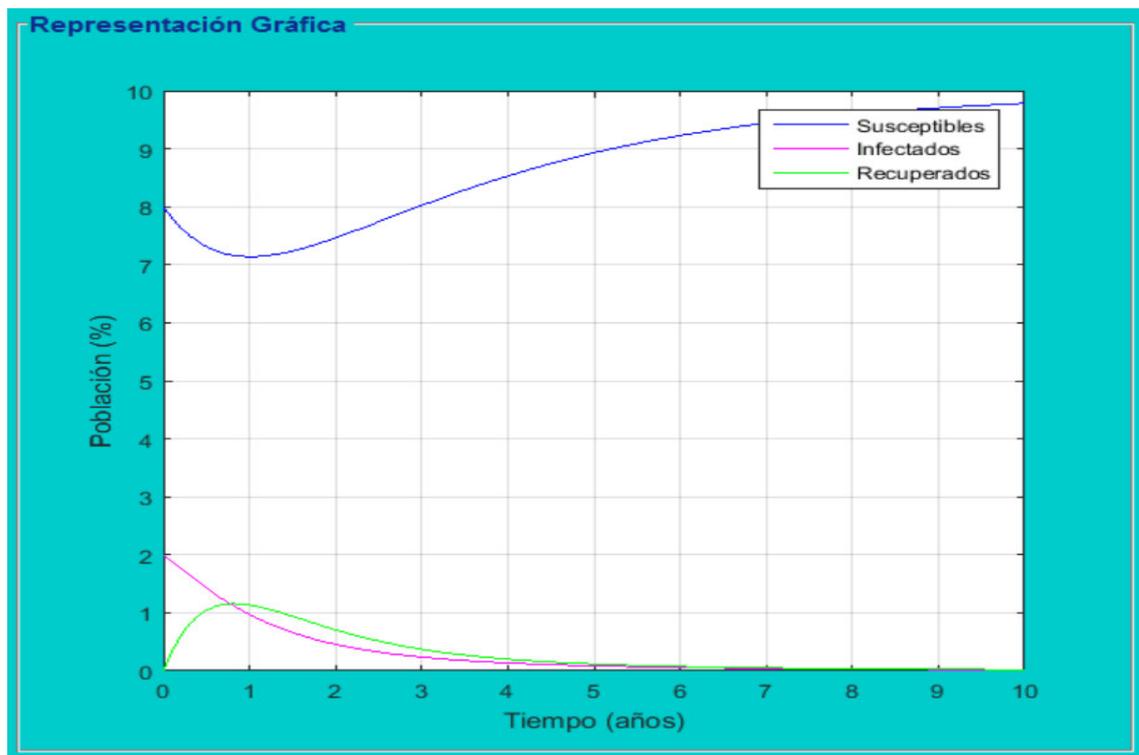
Simulación Computacional

Vamos a utilizar MatLab GUIDE que es un entorno de programación gráfica que ofrece Matlab para poder realizar y ejecutar programas de simulación matemática de forma simple. En este entorno ingresaremos las condiciones iniciales correspondientes a las ecuaciones diferenciales ordinarias, las poblaciones, los intervalos de tiempo y la tolerancia. El programa desarrollado en Matlab se encuentra en el Apéndice.



- a) Para el primer caso vamos a trabajar con una población susceptible donde el 20% de la población total esta infectada, es decir, $S(0) = 0.8$, $I(0) = 0.2$ y $R(0) = 0$.

Los parámetros que usaré cabe recalcar serán los mismos que en el Diagrama de Fases del caso (a). Entonces como la población es constante la tasa de natalidad y mortalidad son iguales ($\mu = 0.5$), la tasa de infección ($\beta = 0.2$) y una tasa de recuperación ($\alpha = 1.7$) y $\gamma = 1.2$, entonces el número básico de reproducción sería $R_0 = 0.90 < 1$.

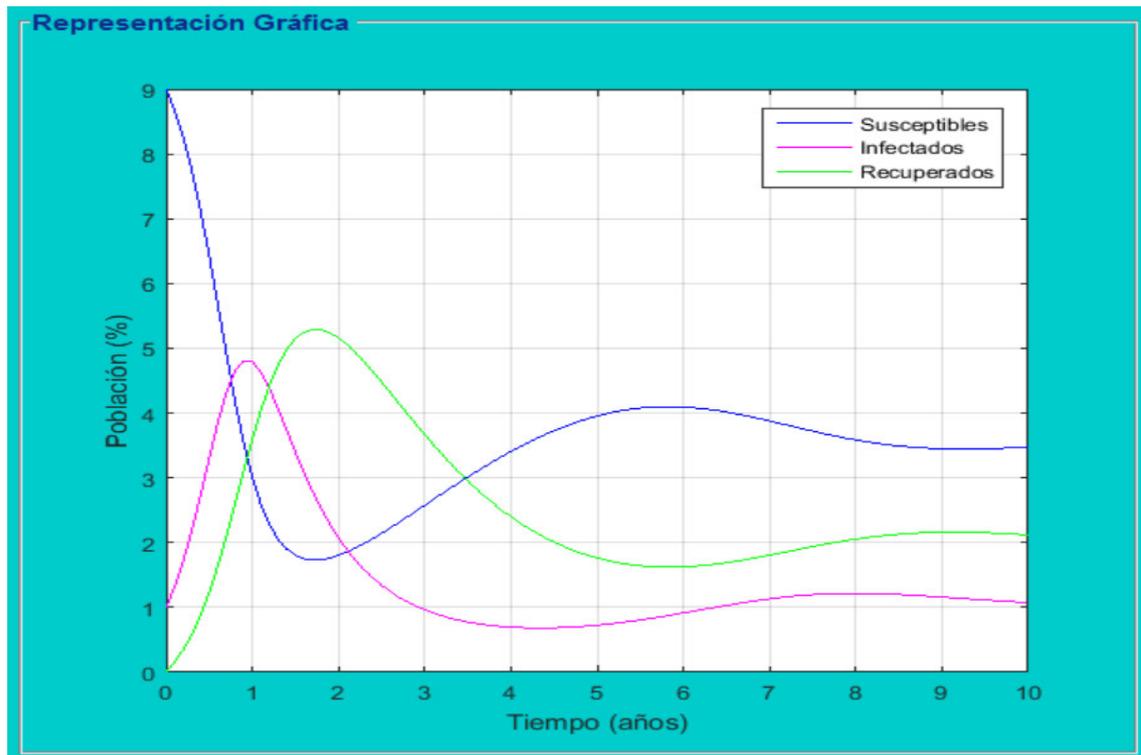


La representación gráfica nos muestra que la población infectada tiende a decrecer con el tiempo, mientras la población susceptible tiende a crecer con el tiempo, de esto podemos deducir que la epidemia se extinguirá, es decir la población infectada se recuperará de la infección endémica ingresando al estado de susceptible.

Si realizamos un cálculo de R_0 nos resulta menor a la unidad, y como ya se ha explicado, cuando esto sucede la epidemia no se propaga y este resultado demuestra lo sustentado anteriormente. Además la población de recuperados y la población de infectados tienden a cero pero no quiere decir que la enfermedad no exista, sino que la enfermedad se ha controlado.

- b) Para el segundo caso vamos a trabajar con una población susceptible donde el 10% de la población total esta infectada, es decir, $S(0) = 0.9$, $I(0) = 0.1$ y $R(0) = 0$.

Los parámetros que usaré cabe recalcar serán los mismos que los del diagrama fase del caso (b). Entonces decimos como la población es constante la tasa de natalidad y mortalidad son iguales ($\mu = 0.3$), la tasa de infección ($\beta = 0.5$) y una tasa de recuperación ($\alpha = 1.5$) y $\gamma = 0.5$, entonces el número básico de reproducción sería $R_0 = 2.7 > 1$.



La representación gráfica nos muestra que cuando la población infectada aumenta entonces la población susceptible disminuye. Es decir existe un incremento potencial de individuos que pasan del estado susceptible al estado infeccioso. Si realizamos un cálculo de R_0 nos resulta mayor a la unidad, y como ya se ha explicado, cuando esto sucede la infección epidémica se propaga y podría producirse una epidemia si no se trata oportunamente a la población.

Conclusiones

- Los modelos matemáticos en general son una herramienta importante para el planteamiento, análisis y prevención de enfermedades. En este trabajo de investigación se presentó un modelo matemático epidemiológico para la dinámica de la malaria, tanto a nivel teórico como también en casos reales. Mostrando la realización de simulaciones porque son de gran utilidad para la prevención y programas de control, que ayuden a determinar cual es la alternativa más correcta según la situación en el que se encuentre la población en estudio.
- El modelo esta conformado por Ecuaciones Diferenciales Ordinarias y este tiene que cumplir ciertos requisitos como existencia, unicidad, estabilidad por lo menos localmente, esto es fundamental en el desarrollo de la dinámica del modelo matemático.
- Cuando utilizamos el modelo requerimos de ciertos parámetros, estos son datos importantes para que el modelo matemático sea lo más realista y por ello se deben utilizar métodos estadísticos óptimos.
- También podemos concluir que mientras los valores paramétricos sean más precisos entonces el modelo del sistema dinámico sera aún más exacto, esto permite una perspectiva más general de la situación de una determinada población.
- El número básico de reproducción es un dato muy importante para la epidemiología. si $R_0 < 1$ la epidemia desaparecerá y si $R_0 > 1$ puede que estemos en un caso de endemia.

Apéndice

Código en MatLab

```
function varargout = InterfazMalaria(varargin)
% INTERFAZMALARIA MATLAB code for InterfazMalaria.fig
%     INTERFAZMALARIA, by itself, creates a new INTERFAZMALARIA or
%     raises the existing
%     singleton*.
%
%     H = INTERFAZMALARIA returns the handle to a new
%     INTERFAZMALARIA or the handle to
%     the existing singleton*.
%
%     INTERFAZMALARIA('CALLBACK', hObject, eventData, handles,...)
%     calls the local
%     function named CALLBACK in INTERFAZMALARIA.M with the given
%     input arguments.
%
%     INTERFAZMALARIA('Property','Value',...) creates a new
%     INTERFAZMALARIA or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before InterfazMalaria_OpeningFcn gets
%     called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to InterfazMalaria_OpeningFcn via
%     varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help InterfazMalaria

% Last Modified by GUIDE v2.5 30-Nov-2016 19:52:43

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
```

```

        'gui_OpeningFcn', @InterfazMalaria_OpeningFcn, ...
        'gui_OutputFcn', @InterfazMalaria_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before InterfazMalaria is made visible.
function InterfazMalaria_OpeningFcn(hObject, eventdata, handles,
    varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to InterfazMalaria (see VARARGIN)

% Choose default command line output for InterfazMalaria
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes InterfazMalaria wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = InterfazMalaria_OutputFcn(hObject, eventdata,
    handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
f1=get(handles.edit1,'string');
f2=get(handles.edit2,'string');
f3=get(handles.edit3,'string');
f=inline(f1,'x','S','I','R');
g=inline(f2,'x','S','I','R');
p=inline(f3,'x','S','I','R');

```

```

y10=str2double(get(handles.edit4,'string'));
y20=str2double(get(handles.edit5,'string'));
y30=str2double(get(handles.edit6,'string'));
x0=str2double(get(handles.edit7,'string'));
xx=str2double(get(handles.edit8,'string'));
h=str2double(get(handles.edit9,'string'));
n=(xx-x0)/h;
q=zeros(1,n);
d1=zeros(1,n);
d2=zeros(1,n);
d3=zeros(1,n);
i=0;
x=x0; w1=y10; w2=y20; w3=y30;

for i=1:n

    k11=h*f(x,w1,w2,w3);
    k12=h*f(x+h/2,w1+k11/2,w2+k11/2,w3+k11/2);
    k13=h*f(x+h/2,w1+k12/2,w2+k12/2,w3+k12/2);
    k14=h*f(x+h,w1+k13,w2+k13,w3+k13);

    k21=h*g(x,w1,w2,w3);
    k22=h*g(x+h/2,w1+k21/2,w2+k21/2,w3+k21/2);
    k23=h*g(x+h/2,w1+k22/2,w2+k22/2,w3+k22/2);
    k24=h*g(x+h,w1+k23,w2+k23,w3+k23);

    k31=h*p(x,w1,w2,w3);
    k32=h*p(x+h/2,w1+k31/2,w2+k31/2,w3+k31/2);
    k33=h*p(x+h/2,w1+k32/2,w2+k32/2,w3+k32/2);
    k34=h*p(x+h,w1+k33,w2+k33,w3+k33);

    w1=w1+(k11+2*k12+2*k13+k14)/6;
    w2=w2+(k21+2*k22+2*k23+k24)/6;
    w3=w3+(k31+2*k32+2*k33+k34)/6;

    x=x0+i*h;
    q(1,i)=x;
    d1(1,i)=w1;
    d2(1,i)=w2;
    d3(1,i)=w3;
end

q=[x0 q];
d1=[y10 d1];
d2=[y20 d2];
d3=[y30 d3];

h2=subplot(1,1,1,'Parent',handles.uipanel4)
plot(h2,q,d1,'b');
plot(q,d1,'b');
hold on;
plot(q,d2,'m');
plot(q,d3,'g-');
xlabel('Tiempo (anos)');
ylabel('Poblaci\on (%)');

```

```

grid;
legend('Susceptibles','Infectados','Recuperados');

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
%        as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
%        as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text

```

```

%          str2double(get(hObject,'String')) returns contents of edit3
%          as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%           called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4
%        as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%           called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5
%        as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%           called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%       str2double(get(hObject,'String')) returns contents of edit6
%       as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%            called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%       str2double(get(hObject,'String')) returns contents of edit7
%       as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%            called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to edit8 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of edit8
%         as a double

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
%             called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9
%         as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
%             called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Bibliografía

- [1] OMS. *Informe Mundial 2015 sobre El Paludismo*.
[Publicado 2016. pp.32]

- [2] MINISTERIO DE LA SALUD. *Boletín Epidemiológico(Lima-Perú)*
[Publicado 28 de febrero del 2016. pp. 158-160.]

- [3] JAVIER VARGAS HERRERA.
Prevención y control de la Malaria y otras enfermedades en el Perú

- [4] IOSR JOURNAL OF MATHEMATICS.
Mathematical Modelling of Malaria Transmission in North Senatorial Zone of Taraba State Nigeria.
[ISSN: 2278-5728. Volume 3, Issue 3 (Sep-Oct. 2012), PP 07-13]

- [6] JOEL M. ADDAWE, JOSE ERNIE C. LOPEZ.
Analysis of age-structured malaria.
[University of the Philippines Baguio. pp.169-180. Vol.5 Nro2. 2012]

- [6] LÓPEZ CRUZ, ROXANA.
Structured SI Epidemic Models with Applications To HIV Epidemic.
[Arizona State University. pp.27-35. 2006.]

- [7] VINCENZO CAPASSO. *Mathematical Structures of Epidemic Systems*
[Springer 2008]