

# Modelos matemáticos para epidemias

Pablo Rodríguez, Diego Sánchez, Jorge Marcos y Jorge Santiago

Mayo 2023

## Resumen

Los modelos matemáticos epidemiológicos nos permiten tomar decisiones informadas y diseñar estrategias efectivas de prevención y control durante una epidemia en base a ciertos parámetros. Con los años, estos modelos han evolucionado para ser cada vez más realistas, aumentando su complejidad. En este trabajo se estudian varios modelos sencillos: SIR, SIRS y SIRV, siendo los dos primeros los modelos clásicos más antiguos. Para el modelo SIR encontramos un conjunto infinito de puntos críticos, de los que solo uno representa la tendencia del sistema para unas condiciones dadas. En cambio, el modelo SIRS sí soporta la endemia y presenta dos puntos críticos gobernados por una bifurcación transcítica en función del parámetro  $R_0$ . Debido a su simplicidad, estos modelos no logran ajustarse correctamente a datos epidemiológicos reales. Mediante el modelo SIRV se demuestra que la vacunación permite disminuir el número de infectados, aunque de forma distinta según la dinámica de vacunación. Finalmente, encontramos modelos más sofisticados, como el SEIR o el SIDARTHE, que logran ajustarse con más fidelidad a la realidad a cambio de aumentar notablemente la complejidad de su estudio.

## 1. Introducción

Durante miles de años, las epidemias han supuesto un problema recurrente que han afectado gravemente a diferentes poblaciones en todo el mundo, y con ello han tenido el poder de alterar notablemente nuestra historia. En un intento por comprender y controlar la propagación de estas enfermedades, se ha recurrido a modelos matemáticos para predecir y analizar su comportamiento. Con los años, estos modelos han evolucionado y se han vuelto cada vez más sofisticados, permitiendo a los expertos en salud pública tomar decisiones informadas y diseñar estrategias efectivas de prevención y control. Podemos subdividir las enfermedades infecciosas en dos categorías: «microparasíticas», asociadas a virus y bacterias; y «macroparasíticas», asociadas a otros parásitos, como protistas y helmintos. La diferencia principal radica en el modo de transmisión, donde los «microparásitos» se transmiten directamente entre hospedadores, mientras que los «macroparásitos» presentan ciclos vitales más complejos, a menudo implicando a hospedadores secundarios [1]. Por su simplicidad, en este trabajo se evaluarán varios modelos matemáticos para la transmisión de enfermedades microparasíticas.

El concepto de una enfermedad infecciosa recuerda al de la depredación, ya que se basa en la interacción de dos organismos que resulta beneficiosa para uno de ellos y perjudicial para el otro. Por tanto, una aproximación válida podría ser emplear como marco el modelo de depredación de Lotka-Volterra, representado en (1), para obtener un modelo de parasitismo modificando las premisas iniciales.

$$\begin{cases} x'(t) = \alpha x - g(y)x \\ y'(t) = \beta xy - \gamma y \end{cases} \quad (1)$$

Sin embargo, encontramos varias razones por la que, matemáticamente, resulta inadecuado analizar una enfermedad como un proceso de depredación. En primer lugar, el número total de patógenos en una población afectada por una enfermedad resulta casi imposible de estimar y supera al de hospedadores por varios órdenes de magnitud. Por otro lado, en cierto modo esta cantidad resulta irrelevante, puesto que es la distribución de los patógenos entre los distintos hospedadores lo que resulta de importancia a la hora de determinar la evolución de la propagación de la enfermedad, y no el número total de patógenos. Además, el modelo de depredación de Lotka-Volterra supone que ambas especies deambulan libremente en el ambiente, por lo que la interacción entre ellas viene dada por una ley de acción de masas (término  $\beta xy$ ). Sin embargo, las enfermedades microparasíticas suelen transmitirse por contacto o proximidad entre individuos hospedadores.

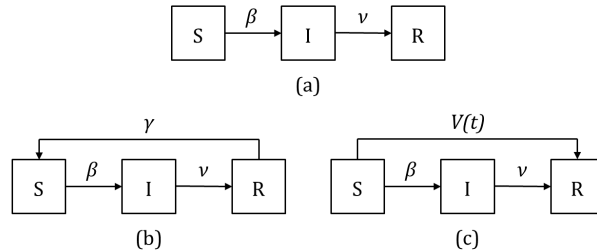
En cambio, la mayoría de modelos epidemiológicos se basan en la subdivisión de la población en distintas clases según su salud. De esta forma, se deja de considerar directamente a la población de parásitos o patógenos para trabajar únicamente con poblaciones de hospedadores. Los primeros modelos estudiados consideraban una división de la población de hospedadores en individuos susceptibles ( $S$ ), que pueden contraer la enfermedad; individuos infecciosos ( $I$ ), que padecen la enfermedad y pueden transmitirla; e individuos recuperados o retirados ( $R$ ), que no pueden contraer la enfermedad, bien porque han desarrollado una inmunidad contra ella o porque han fallecido a causa de ella. Estas subdivisiones quedan recogidas en el denominado modelo SIR, propuesto por W. O. Kermack y A. G. McKendrick en 1927 [2]. Las ecuaciones que propusieron Kermack y MacKendrick para este modelo epidemiológico fueron:

$$\begin{cases} S'(t) = -\beta IS \\ I'(t) = \beta IS - \nu I \\ R'(t) = \nu I \end{cases} \quad (2)$$

donde se asume que el nivel de transmisión de la enfermedad es proporcional, con tasa  $\beta$ , al número medio de encuentros entre individuos susceptibles e individuos infecciosos, dado por una ley de acción de masas ( $I \cdot S$ ). De esta forma,  $S(t)$  y  $I(t)$  disminuyen y aumentan, respectivamente, con el tiempo conforme a esta interacción. Por otro lado, la población  $I(t)$  decae exponencialmente, con tasa  $\nu$ , a causa de la muerte o recuperación de los individuos infectados.

Con los años, se han propuesto múltiples variaciones de este modelo, por ejemplo, considerando pérdida de memoria inmunológica (SIRS), dinámicas vitales o ausencia de inmunidad (SIS). Distintas combinaciones de estos factores dan lugar a modelos distintos, más o menos realistas según el caso. A continuación se estudiará el modelo SIR básico Kermack-McKendrick de 1927, así como dos variaciones: una considerando un proceso de vacunación y otra considerando una pérdida de la memoria inmunológica. A la primera variación se le referirá como modelo SIRV, y a la segunda como modelo SIRS. Nótese que en la literatura en ocasiones se le denomina SIRV al modelo SIRS. La principal diferencia se encuentra en que en el modelo SIRV (con vacunación) se produce un "movimiento" de la población susceptible a la recuperada sin pasar por la infección, mientras que en el modelo SIRS se produce el movimiento contrario a causa de la pérdida de inmunidad.

A este tipo de modelos continuos se les denomina «modelos compartimentales», ya que consideran un flujo ordenado entre distintos compartimentos que subdividen la población total de hospedadores. Esto difiere conceptualmente de otros modelos, como el modelo de depredación Lotka-Volterra, donde no hay conversión entre especies, sino que nacen o mueren según su interacción. En cambio, en los modelos compartimentales sí se produce esta conversión, de modo que una disminución en un compartimento se traduce en el equivalente aumento de otro cuando el modelo es cerrado, lo cual se observa fácilmente en el sistema de ecuaciones. En la **Figura 1** se muestra una representación de este sistema compartimental para cada uno de los tres modelos mencionados. Por otro lado, ninguno de estos modelos considerará dinámicas vitales, por lo que en todos ellos se cumple  $N = S + I + R$ , donde  $N$  es el número total de individuos. Esto queda plasmado formalmente en (16).



**Figura 1:** Modelos compartimentales tipo SIR estudiados en este trabajo. (a) Modelo SIR básico; (b) Modelo SIRS; (c) Modelo SIRV.

Todos estos modelos comparten los parámetros  $\beta$  y  $\nu$ , que conservan su significado. El parámetro  $\beta$  representa la probabilidad de que una interacción entre un individuo susceptible ( $S$ ) y uno infectado ( $I$ ) resulte en una transmisión exitosa, mientras que  $\nu$  representa la tasa con la que se curan los infectados, de forma que  $1/\nu$  representa el período medio de infectividad.

En estos modelos, el ratio  $\beta/\nu$  tiene una importante interpretación, pues representa la fracción de individuos que han entrado en contacto con un infectado durante su período de infectividad y que, en consecuencia, se han contagiado. Así, definimos

$R_0 = (N\beta)/\nu$  (a veces denotado como  $\sigma$ ) como el *infectious contact number* o *basic reproduction number*, es decir, el número medio de susceptibles contagiados por un solo infectado a lo largo de su período de infectividad [3].  $R_0$  resulta de gran relevancia en lo que concierne la estabilidad de los sistemas, especialmente en el SIRS.

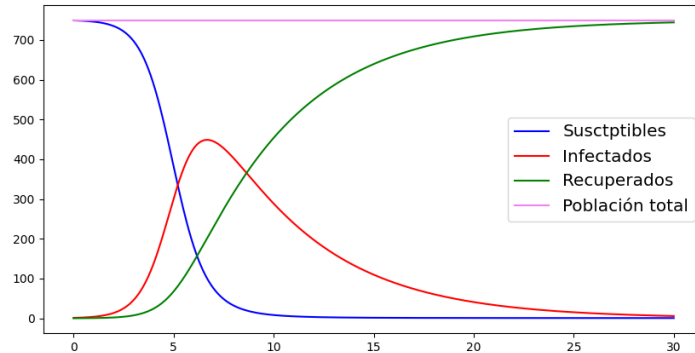
## 2. Modelos SIR

### 2.1. Modelo básico SIR

Este modelo es una de los más simples posibles, puesto que no considera ningún fenómeno salvo el del contagio y la cura o muerte. En este modelo se produce un flujo continuo unidireccional desde  $S$  a  $I$  y de  $I$  a  $R$ . Al ser un sistema cerrado (sin dinámicas vitales, emigración ni inmigración), la población total ( $N$ ) se mantiene constante (16). Para estudiar la estabilidad del SIR, podemos empezar por calcular los puntos críticos o de equilibrio del sistema (2), en los cuales el sistema no evoluciona. El desarrollo se incluye en (17). Vemos que este sistema presenta un conjunto infinito de puntos de equilibrio, que podríamos definir como el espacio vectorial:

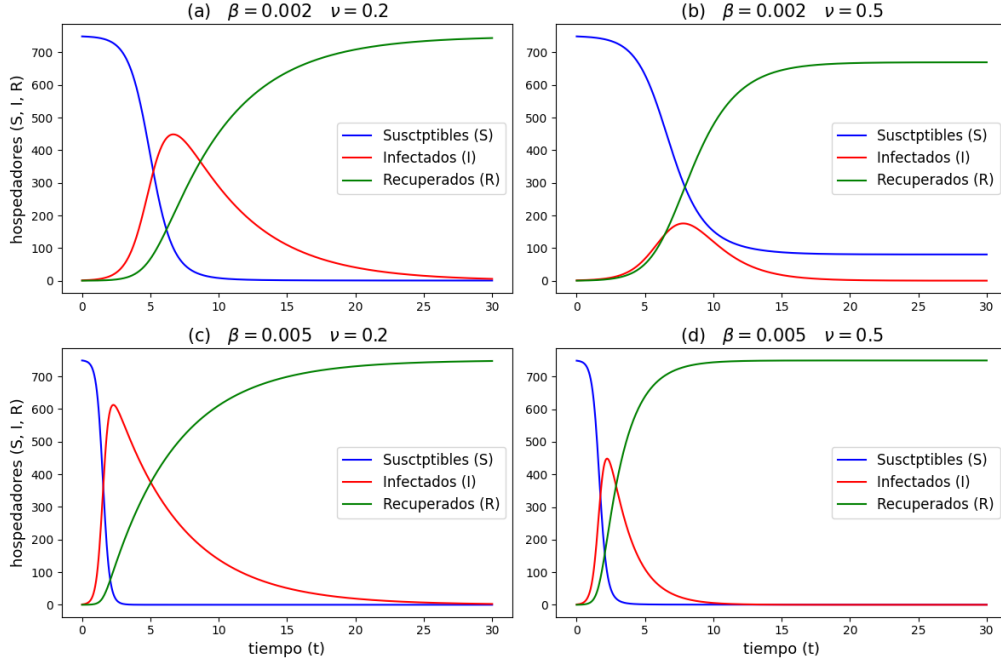
$$P\{(S, 0, N - S) \mid S, N \in \mathbb{R}^+ \mid S \leq N\} \quad (3)$$

donde sólo consideramos soluciones con sentido biológico, es decir, poblaciones con valores positivos. Este resultado es, en realidad, sumamente razonable. En una población sin infectados, no se produce infección ni epidemia. Es decir, los susceptibles no se contagian, por lo que no tienen forma de adquirir inmunidad o morir por la enfermedad. De algún modo, estamos «cortando» el flujo entre los compartimentos del sistema. En cambio, para valores positivos de  $I$ , la infección evoluciona hacia uno, y solo uno, de los puntos en el espacio  $P$ . Es decir, de todos los puntos de  $P$ , sólo uno se comporta como asintóticamente estable para un determinado PVI (problema de valor inicial). En la **Figura 2** se muestra gráficamente este comportamiento. El resultado es el esperado: los susceptibles se contagian y los infectados se curan o fallecen. Como resultado, se produce un pico de infección que se desvanece con el tiempo, de forma que el sistema evoluciona hacia un punto de equilibrio.



**Figura 2:** Simulación numérica del modelo SIR para unas poblaciones iniciales  $P_0 = (749, 1, 0)$  y parámetros  $\nu = 0,2$  y  $\beta = 0,002$

Como bien podría esperarse, los parámetros  $\beta$  y  $\nu$  influyen en la evolución del sistema (**Figura 3**). Al aumentar  $\beta$ , con  $\nu$  constante, el pico de infección aumenta de forma más pronunciada y alcanza un mayor máximo, pero conserva una «cola» a la derecha. Es decir, los infectados aumentan más rápidamente pero siguen curándose o muriéndose a la misma velocidad. En cambio, al aumentar  $\nu$ , con  $\beta$  constante, el pico se hace más bajo y estrecho. Es decir, al disminuir el periodo de infectividad, la epidemia es menos severa y se resuelve antes.



**Figura 3:** Simulación SIR para distintos valores de  $\nu$  y  $\beta$ . (a) corresponde al mismo caso que la Figura 2. Poblaciones iniciales (749,1,0).

En la gráfica (b) de la **Figura 3** observamos un fenómeno interesante, en el que una fracción de susceptibles nunca llega a infectarse (en este caso, más de una décima parte). Es decir, aunque pudiera parecer que en todo caso el sistema evoluciona hacia el punto  $(0, 0, N)$ , vemos que no es así. De hecho, como se verá más adelante, éste punto representa la tendencia para valores muy altos de  $\beta$  (próximos a 1) y muy bajos de  $\nu$  (próximos a 0). En cualquier caso, este claro contraejemplo nos revela que, en el resto de gráficas, la población de susceptibles no tiende a 0, sino a un valor sencillamente pequeño. Por ejemplo, en la **Figura 2**,  $S$  tiende aproximadamente a 0,416 (véase **cuaderno 1**). Además, evidentemente esto demuestra que podemos obtener resultados distintos en función de las condiciones iniciales.

Nótese que esta población final de susceptibles, denotada como  $S_f$ , podría interpretarse como los supervivientes a la epidemia en caso de la enfermedad fuera siempre letal. Sin embargo, éstas son simples suposiciones adicionales que no están incluidas en la definición del modelo SIR. Para poder hablar propiamente de «supervivientes», se necesita elaborar un modelo que distinga entre curados y fallecidos, por ejemplo,

partiendo  $\nu$  en dos parámetros independientes  $\omega$  (tasa de cura) y  $\delta$  (tasa de muerte), que pueden ser o no constantes. Esta variación no será abordada en este trabajo.

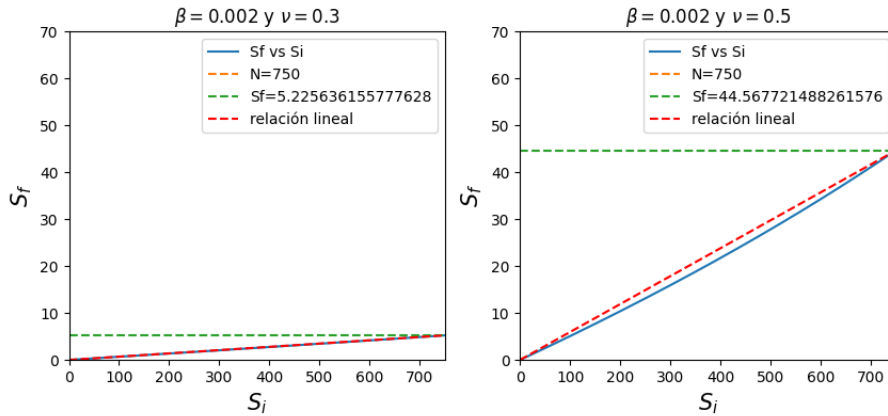
En un intento por predecir qué punto concreto del espacio  $P$  representa la tendencia del sistema para un PVI dado, podemos estudiar la estabilidad de estos puntos críticos mediante la aproximación lineal con la matriz jacobiana del sistema [4]. Para facilitar los cálculos podemos realizar una reducción dimensional. En este caso, se eliminó  $R$  aprovechando  $R = N - S - I$ . Con ello, se calculó la matriz jacobiana y se evaluó en los puntos críticos del espacio  $P$  (4). Además, se calculó la traza y el determinante de la matriz evaluada (5).

$$J = \begin{pmatrix} -I\beta & -S\beta \\ I\beta & S\beta - \nu \end{pmatrix} \quad J(S, 0) = J^* = \begin{pmatrix} 0 & -S\beta \\ 0 & S\beta - \nu \end{pmatrix} \quad (4)$$

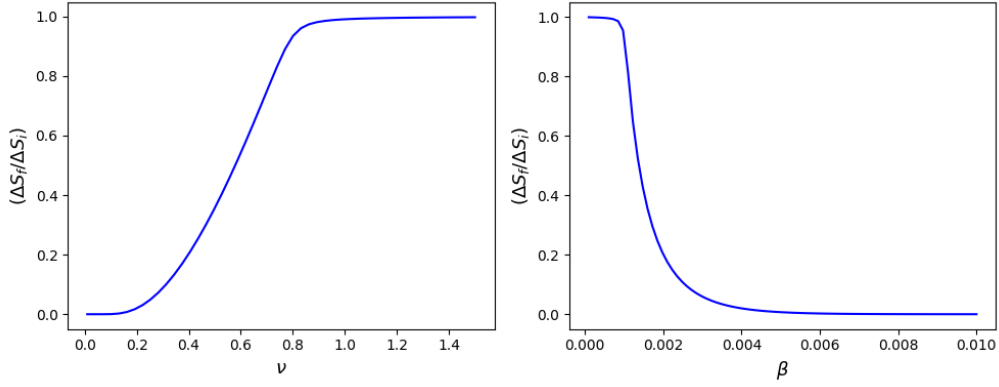
$$tr(J^*) = (S\beta - \nu) \quad det(J^*) = 0 \quad (5)$$

Debido a que el determinante de la matriz es nulo para cualquier punto crítico del sistema, no podemos evaluar la estabilidad de ninguno de ellos mediante este método. Para lograr obtener información sobre su estabilidad se deberá aplicar métodos más sofisticados, como, por ejemplo, los empleados en un estudio de 2009 de Chalub y Souza [5], donde se presenta una derivación del modelo SIR mediante una aproximación de campo medio desde una versión discreta del modelo. Sin embargo, más allá de obtener la relación real entre los parámetros del sistema y la estabilidad de los puntos críticos, a continuación se muestran algunos hallazgos con respecto a estas relaciones, aunque de forma meramente aproximada.

En primer lugar, se encontró una relación lineal entre la población inicial ( $S_i$ ) y final de susceptibles ( $S_f$ ) a la que tiende el sistema (**Figura 4**), siempre que la suma  $S_i + I_i$  se mantenga constante. Nótese cómo la población inicial de recuperados ( $R_i$ ) no influye en el valor de  $S_f$ , ya que los recuperados no afectan al sistema en general; simplemente representan la «salida» del mismo. Esto puede verse fácilmente con simulaciones numéricas (véase **cuaderno 1**).



**Figura 4:** Relación entre las poblaciones iniciales y finales de susceptibles. Se consideró un gradiente de condiciones iniciales  $P_i \in \{(0, 750, 0) \dots (750, 0, 0)\}$ , tal que  $I_i + S_i = cte$ .



**Figura 5:** Relación entre el ratio  $S_f/S_i$  y los parámetros  $\nu$  y  $\beta$ . Para la primera gráfica se fijó  $\beta = 0,002$ , y para la segunda se fijó  $\nu = 0,4$ .

Se vio también que el ratio  $S_f/S_i$  (que representa la pendiente en la **Figura 4**) es dependiente de  $\nu$  y  $\beta$  (**Figura 5**). Esta dependencia viene dada por curvas sigmoideas, de modo que se observan las siguientes tendencias:

$$\lim_{\frac{\nu}{\beta} \rightarrow \infty} \left( \frac{S_f}{S_i} \right) = 1 \quad \lim_{\frac{\nu}{\beta} \rightarrow 0} \left( \frac{S_f}{S_i} \right) = 0 \quad (6)$$

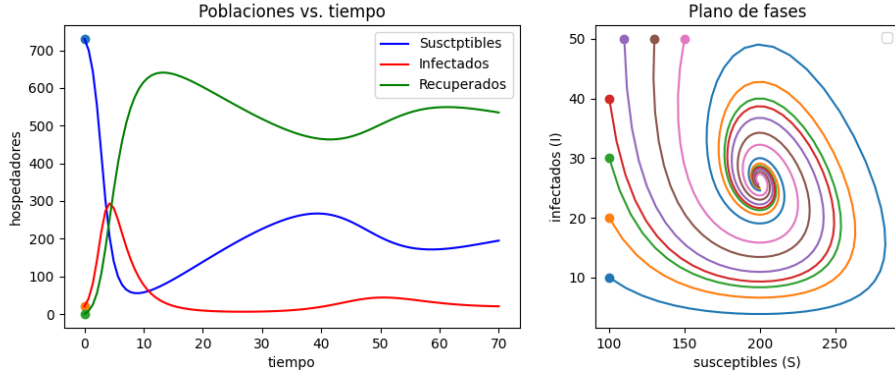
Establecer una relación numérica entre  $S_f/S_i$  y los parámetros  $\nu$  y  $\beta$ , como, por ejemplo,  $S_f = m(\nu, \beta) \cdot S_i$ ; sería suficiente para predecir la tendencia del sistema a partir de cualquier PVI. Esto queda desarrollado en (18)

## 2.2. Modelo SIRS

Se trata de un caso más general, en el que se produce una pérdida de memoria inmunológica a tasa  $\gamma$ , como se ve en (7). Por tanto, el SIR queda incluido dentro de este modelo como un caso específico para  $\gamma = 0$ . Las ecuaciones propuestas por Bailey en 1975 para este modelo [6] fueron:

$$\begin{cases} S'(t) = -\beta SI + \gamma R \\ I'(t) = \beta SI - \nu I \\ R'(t) = \nu I - \gamma R \end{cases} \quad (7)$$

En el modelo SIRS se produce un movimiento de la población recuperada ( $R$ ) a la susceptible ( $S$ ). En realidad, esto asume que la enfermedad no es mortal en ningún caso. Este movimiento permite la aparición de un fenómeno interesante: la **endemia**. Es decir, la aparición constante de nuevos susceptibles podría permitir la perpetuación de la enfermedad en la población de hospedadores. En la **Figura 6** se observa fácilmente esta tendencia, así como las oscilaciones en los valores de  $S$ ,  $I$  y  $R$ , que recuerdan a las famosas «olas» de infección durante una epidemia.



**Figura 6:** Simulación numérica del modelo SIRS para los parámetros  $\beta = 0,002$ ,  $\nu = 0,4$ ,  $\gamma = 0,02$  y  $N = 750$ . En el gráfico de la izquierda se representan a las tres poblaciones respecto al tiempo, comenzando en el punto  $(730, 20, 0)$ . En el gráfico de la derecha se representa el plano de fases  $S$  vs.  $I$  con varias condiciones iniciales, indicadas como puntos.

### 2.2.1. Análisis cualitativo del modelo SIRS

Estos comportamientos pueden analizarse en detalle mediante un estudio cualitativo de la estabilidad del sistema. Para estudiar la estabilidad podemos empezar por calcular los puntos de equilibrio del sistema, que, tal y como se demuestra en (19) y confirma en el **cuaderno 2**, son:

$$P_{e1} = (N, 0, 0) \quad P_{e2} = \left( \frac{\nu}{\beta}, \frac{\gamma(\beta N - \nu)}{\beta(\gamma + \nu)}, \frac{\nu(\beta N - \nu)}{\beta(\gamma + \nu)} \right) \quad (8)$$

Para estudiar la estabilidad por la aproximación lineal, calculamos la matriz jacobiana. En este caso, para facilitar los cálculos, hacemos una reducción dimensional, pues siempre se cumple que  $N = S + I + R$ , lo que permite eliminar la variable  $R$  sustituyéndola por  $N - S - I$ , dando lugar al sistema (9). Esta reducción de variables no afecta al comportamiento del sistema (véase **cuaderno 2**).

$$\begin{cases} S'(t) = f(S, I) = -\beta SI + \gamma(N - S - I) \\ I'(t) = g(S, I) = \beta SI - \nu I \end{cases} \quad (9)$$

Los resultados del análisis cualitativo realizado con Python se muestran en (11) y (12), donde  $J_1$  y  $J_2$  se obtienen como resultado de evaluar la matriz jacobiana (10) del sistema (9) en  $P_{e1}$  y  $P_{e2}$  (8), respectivamente. En (10),  $(S^*, I^*)$  denota un punto crítico.  $J_1$  y  $J_2$  permiten obtener los parámetros con los que poder evaluar la estabilidad del sistema en  $P_{e1}$  y  $P_{e2}$ , que son los autovalores ( $\lambda_{i1}$  y  $\lambda_{i2}$ ) y los invariantes (traza y determinante).

$$J = \begin{pmatrix} f_S(S^*, I^*) & f_I(S^*, I^*) \\ g_S(S^*, I^*) & g_I(S^*, I^*) \end{pmatrix} = \begin{pmatrix} -\beta I - \gamma & -\beta S - \gamma \\ \beta I & \beta S - \nu \end{pmatrix} \quad (10)$$



### 2.2.2. Estabilidad del primer punto crítico

Los invariantes calculados a partir de la matriz jacobiana evaluada en  $P_{e1}$  se muestran en (11), donde también se ha impuesto la condición de estabilidad asintótica.

$$J_1 = \begin{pmatrix} -\gamma & -N\beta - \gamma \\ 0 & N\beta - \nu \end{pmatrix} \quad \begin{cases} tr(J_1) = N\beta - \gamma - \nu < 0 \\ det(J_1) = \gamma(-N\beta + \nu) > 0 \end{cases} \quad (11)$$

Nótese que  $N\beta$  se puede escribir como  $R_0\nu$ . La estabilidad del modelo depende de los signos del determinante y la traza. Sabiendo que los parámetros del modelo ( $\gamma, \nu, \beta$  y  $N$ ) son positivos, vemos que la estabilidad de este punto crítico depende de  $R_0$ . Si  $R_0 < 1$ , entonces  $tr(A) < 0$  y  $det(A) > 0$ , por lo que el punto sería asintóticamente estable. Si  $R_0 > 1$ , entonces  $det(A) < 0$  y, por tanto, el punto se comporta como inestable. Cuando  $R_0 = 1$ , tenemos que  $tr(A) < 0$  y  $det(A) = 0$ , por lo que no se puede conocer *a priori* la estabilidad de este punto en este caso.

### 2.2.3. Estabilidad del segundo punto crítico

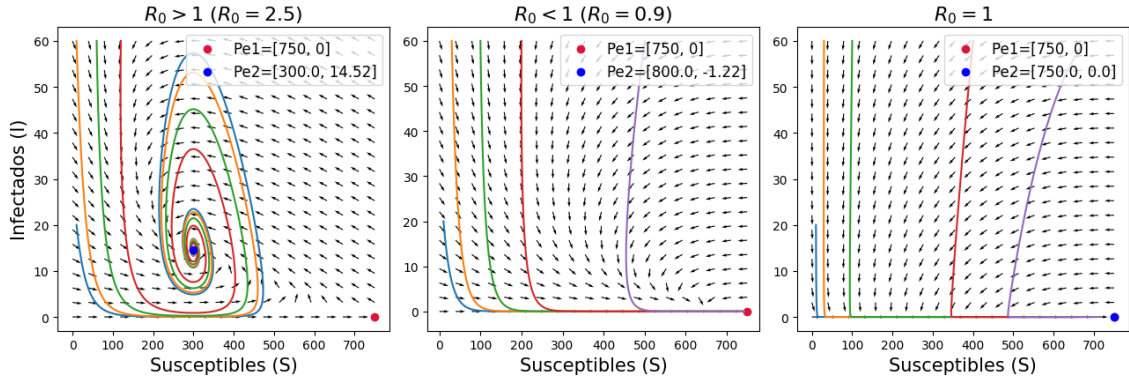
Al igual que en el caso anterior, se evalúa la matriz jacobiana en el  $P_{e2}$  y calculamos sus invariantes. El resultado se muestra en (12), donde se ha impuesto la condición de estabilidad asintótica, es decir,  $tr(J_2) < 0$  y  $det(J_2) > 0$ .

$$J_2 = \begin{pmatrix} \frac{\gamma(-N\beta - \gamma)}{\frac{\gamma + \nu}{\gamma(N\beta - \nu)}} & -\gamma - \nu \\ \frac{\gamma(N\beta - \nu)}{\gamma + \nu} & 0 \end{pmatrix} \quad \begin{cases} tr(J_2) = \frac{\gamma(-N\beta - \gamma)}{\gamma + \nu} < 0 \\ det(J_2) = \gamma(N\beta - \nu) > 0 \end{cases} \quad (12)$$

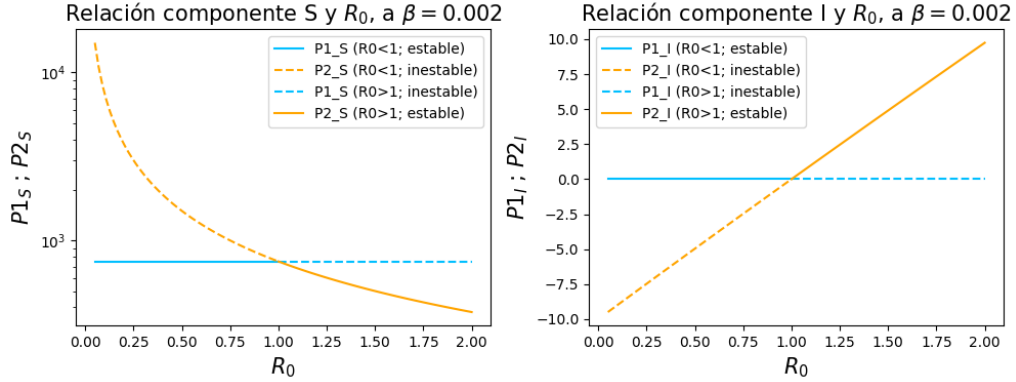
Sabiendo que los parámetros  $N, \beta, \nu$  y  $\gamma$  son positivos, vemos que siempre se cumple  $tr(J_2) < 0$ . Sin embargo, el signo del determinante sí depende de  $R_0$ . Cuando  $R_0 > 1$  vemos que  $det(J_2) > 0$  y el punto es asintóticamente estable. Si  $R_0 < 1$ , entonces  $det(J_2) < 0$  y el punto sería inestable. Si  $R_0 = 1$ , entonces  $det(J_2) = 0$  y por tanto no se puede saber *a priori* su estabilidad.

### 2.2.4. Dinámica de los puntos críticos

En resumen, el sistema presenta dos puntos críticos  $P_{e1}$  y  $P_{e2}$  (8). En concreto,  $P_{e1}$  es asintóticamente estable para  $R_0 < 1$  e inestable para  $R_0 > 1$ , mientras que  $P_{e2}$  es inestable para  $R_0 < 1$  y asintóticamente estable para  $R_0 > 1$ , y no conocemos la estabilidad de ninguno cuando  $R_0 = 1$ . En la **Figura 7** podemos ver el comportamiento del sistema. Se observa cómo  $P_{e2}$  carece de sentido biológico cuando  $R_0 < 1$  y que el sistema tiende a  $(N, 0, 0)$  cuando  $R_0 = 1$  (**cuaderno 2**), pero más interesante es el hecho de que ambos puntos coinciden cuando  $R_0 = 1$ . En ese caso concreto pasamos a tener un único punto crítico, de estabilidad «híbrida». Se trata nada menos que de una **bifurcación transcítica**, en la que ambos puntos críticos se cruzan e intercambian estabilidad al aumentar  $R_0$  en torno a 1 (**Figura 8**). Curiosamente, la componente  $S$  de  $P_{e2}$  aumenta «superexponencialmente» cuando  $R_0$  se aproxima a 0. No hemos podido analizar en profundidad este comportamiento debido a que no hemos logrado que los puntos críticos pasen a depender de  $N, \gamma$  y  $R_0$  y en ningún caso de  $\nu$  y  $\beta$ ; siempre quedaba  $\nu$  o  $\beta$  (20).



**Figura 7:** Plano de fases entre susceptibles ( $S$ ) e infectadas ( $I$ ), para varias condiciones iniciales considerando tres casos:  $R_0 > 1$ ,  $R_0 < 1$  y  $R_0 = 1$ . Se muestra también el campo de direcciones para cada caso



**Figura 8:** Diagrama de bifurcación del modelo SIRS para el parámetro  $R_0$ . Cada gráfica corresponde a una componente,  $S$  e  $I$ , respectivamente, de los puntos críticos. La línea continua indica estabilidad, mientras que la discontinua indica inestabilidad. Primera gráfica con eje  $Y$  logarítmico.

### 2.2.5. Discusión

Tras realizar el estudio cualitativo del modelo y estudiar el comportamiento de sus dos puntos críticos, obtenemos las siguientes conclusiones para el modelo SIRS: Cuando una enfermedad surge en una determinada población, ésta acabará resolviéndose si  $R_0 < 1$ , de modo que los infectados desaparecen ( $I_f = 0$ ) y toda la población vuelve a ser susceptible ( $S_f = N$ ). En cambio, en caso de que esta enfermedad presente  $R_0 > 1$ , entonces adquirirá un comportamiento endémico, de modo que las poblaciones de susceptibles ( $S$ ), infectados ( $I$ ) y recuperados ( $R$ ) tienden hacia unos determinados valores positivos, dados por los parámetros  $N$ ,  $\nu$ ,  $\beta$  y  $\gamma$  del modelo. Además, este modelo presenta una bifurcación transcítica para  $R_0$ , de modo que el sistema presenta un único punto crítico para el caso especial  $R_0 = 1$ , donde, aunque esperaríamos que  $I$  se mantuviera en 1, vemos que, en realidad, tiende a 0 al igual que en el caso  $R_0 < 1$ .

### 2.3. Políticas de vacunación en modelos epidemiológicos

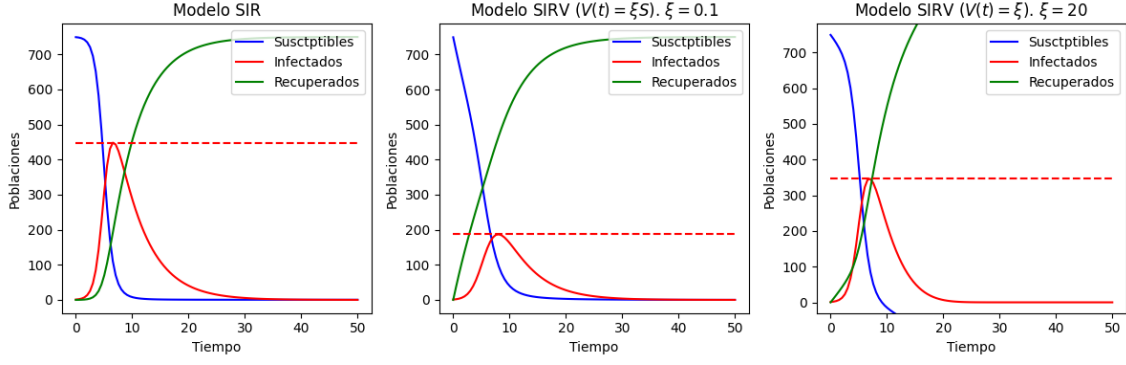
La inmunización es un factor muy importante a tener en cuenta ya que la inmunización, a través de vacunas, es usada para el control de la enfermedad, así como para su tratamiento y eliminación. Esto es debido a que aquellas personas vacunadas adquieren la inmunidad sin pasar por la infección. De esta manera, según las ecuaciones del modelo, las poblaciones infectadas, al partir de las susceptibles, serían menores y la enfermedad se podría erradicar más fácilmente.

Un efecto indirecto de la vacunación es el efecto conocido como «inmunidad de rebaño». El porcentaje de la población vacunada, dependerá altamente de la capacidad infectiva del patógeno causante. Sin embargo, este efecto mencionado anteriormente indica que no es necesaria una vacunación completa de la población, ya que en algunas de las infecciones por virus, solo ha sido necesaria la vacunación de cerca del 85 % de la población [7], para que este virus no se pueda seguir desarrollando, debido a que la población no posee la susceptibilidad necesaria. Esto supone una ventaja ya que los efectos secundarios que puedan ocasionar las vacunas, aparecen en menor medida. De esta manera, para que la «inmunidad de rebaño» tenga un efecto sobre la población, la vacunación debe de haber permitido que el valor de  $R_0$  sea inferior a 1

Varios estudios han implementado la tasa de vacunación  $V(t)$  en los modelos SIR y SIRS (13). Se vacuna a la población susceptible  $S$ , que se convierte en recuperados  $R$ , a una tasa de vacunación  $V(t)$  que varía con el tiempo (depende de la evolución de la enfermedad, entre otros factores). De esta manera, se crean los modelos SIRV y SIRS<sub>V</sub>, regidos por los siguientes sistemas de ecuaciones, respectivamente:

$$\begin{cases} \frac{dS}{dt} = -\beta SI - V(t)S \\ \frac{dI}{dt} = \beta SI - \nu I \\ \frac{dR}{dt} = \nu I + V(t)S \end{cases} \quad \begin{cases} \frac{dS}{dt} = -\beta SI + \gamma R - V(t)S \\ \frac{dI}{dt} = \beta SI - \nu I \\ \frac{dR}{dt} = \nu I - \gamma R + V(t)S \end{cases} \quad (13)$$

En este trabajo hemos considerado dos hipótesis para la tasa de vacunación:  $V_1(t) = \xi$  y  $V_2(t) = \xi S$ . La primera se basa más en la dinámica real, donde cada día se vacuna a un número constante de personas, independientemente de cuántas personas queden sin vacunar. La segunda hipótesis simplifica el modelo, pues nos proporciona escenarios con sentido biológico, aunque menos lógicos, al contrario que en la primera hipótesis. El efecto de ambas hipótesis en el SIR se representa en la **Figura 9**. Con el tiempo, la primera hipótesis genera valores negativos de susceptibles, y valores de recuperados por encima de  $N$ .

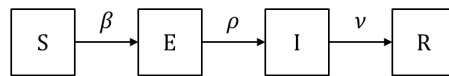


**Figura 9:** Comparación de la dinámica de poblaciones en los modelos SIR y SIRV con dos hipótesis, con tasa de vacunación  $\xi = 0, 1$  y  $\xi = 20$ , respectivamente. Se tomaron valores iguales para el resto de los parámetros:  $\beta = 0,002, \nu = 0,2, N = 750$  y  $P_i = (749, 1, 0)$ . En la segunda gráfica (hipótesis  $V(t) = \xi S$ ), la vacunación depende de  $S$ , mientras que en la tercera (hipótesis  $V(t) = \xi$ ) la vacunación es constante.

Se observa una importante disminución del pico de infectados, además de un aumento de personas recuperadas en menor tiempo. Los efectos de la vacunación en la dinámica del SIRS son semejantes (véase **cuaderno 3**). En la tercera gráfica se pierde sentido biológico dado que la población  $S$  se hace menor que 0. Con esto, se resalta la importancia de la vacunación en la dinámica de los modelos y su impacto positivo en el control de la enfermedad.

### 3. Modelo SEIR

A la vista de los resultados anteriores, el modelo SIR no se adapta bien a un modelo capaz de predecir o analizar las variaciones que pueden tener lugar en una epidemia, ya que por ejemplo no tiene en cuenta la presencia de personas infectadas, que no transmiten el agente patógeno, o simplemente personas que están incubando el virus. El modelo epidemiológico SEIR, propone añadir una nueva población  $E(t)$ , la cual es una población infectada que no muestra síntomas (periodo de incubación), y, a diferencia de la población infectada  $I(t)$ , no puede infectar. Sin embargo, tampoco puede "volver a infectarse", por lo que, al considerar esta subpoblación, disminuimos el número de interacciones efectivas entre susceptibles e infectados. Dado que se incluye una nueva población, se debe incluir un nuevo parámetro que nos permite obtener dichos individuos, a partir de una población susceptible. Este parámetro es  $\rho$ , de tal forma que  $1/\rho$  es el tiempo de incubación, quedando:



**Figura 10:** Representación compartimental del modelo SEIR.

Como las ecuaciones del modelo SEIR derivan de las ecuaciones proporcionadas por el modelo SIR, añadiendo la nueva población  $E(t)$ , así como el nuevo parámetro  $\rho$ , las ecuaciones que rigen el modelo quedarían:

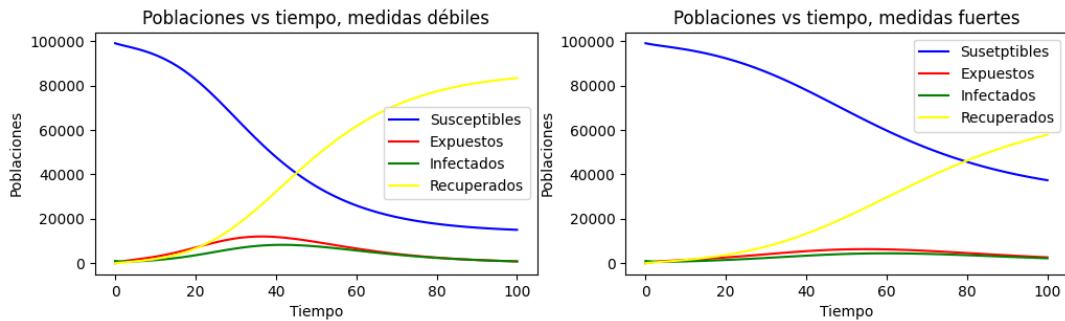
$$\begin{cases} S'(t) = -\beta SI \\ E'(t) = \beta SI - \rho E \\ I'(t) = \rho E - \nu I \\ R'(t) = \nu I \end{cases} \quad (14)$$

Una vez tenemos el sistema de ecuaciones, se puede realizar un análisis del modelo. Debido al mayor número de ecuaciones, no se realizará un estudio cualitativo del SEIR. Sin embargo, puede verse fácilmente cómo, al igual que en el SIR, existen infinitos puntos críticos del tipo  $(S, 0, 0, N - S)$ , donde  $S, N \in \mathbf{R}^+$  y  $S \leq N$ .

Podemos aplicar este modelo a la pandemia del COVID-19 empleando valores de los parámetros  $\beta$ ,  $\rho$ ,  $\nu$  calculados a partir de datos reales, concretamente:  $\beta = 1$ ,  $\rho = 1.7$ , y  $\nu = 1.5$ . No obstante, el parámetro  $\beta$  puede no ser constante, si, por ejemplo, se toman medidas para reducir la tasa de infección (aislamiento, uso perceptivo de mascarilla, etc. ) y la población  $N$  las cumple. De esta forma, se han desarrollado estudios con parámetros  $\beta$  que varían con el tiempo [8], por ejemplo:

$$\beta(t) = \beta_0 \cdot (1 - \alpha(t)) \cdot (1 - \frac{D(t)}{N})^k \quad (15)$$

donde  $\beta_0$  es la tasa de infección sin medidas de contención y  $\alpha(t)$  corresponde a una función que toma valores entre 0 y 1, y representa el efecto de las acciones gubernamentales. Cuanto mayor impacto tengan estas medidas, menor será el término  $(1 - \alpha(t))$ , y la tasa de infección disminuye. Por otro lado,  $D(t)$  representa la sensación de riesgo de la población, debido a las muertes o casos críticos producidos durante la epidemia. El factor  $k$  representa el nivel de acción de los individuos. Esto da una idea de que, cuanto mayor sea la preocupación de la población y mayor sea su reacción ante la situación, la gente se aislará voluntariamente y el término  $(1 - (D(t)/N))^k$  disminuirá, haciéndolo así también la tasa de infección. Es decir, la tasa de infección se verá disminuida según las medidas de contención que se impongan y la actitud de la población (**Figura 11**) (**cuaderno 4**).

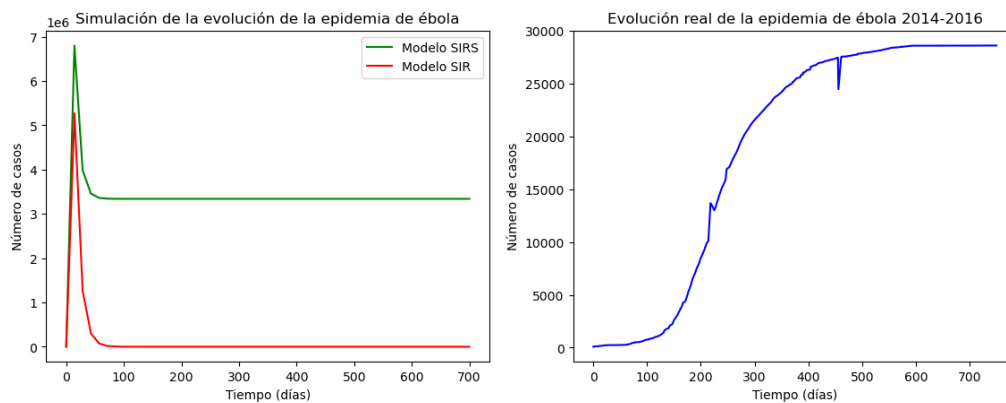


**Figura 11:** Evolución de las distintas poblaciones a lo largo del tiempo, en presencia de medidas gubernamentales débiles ( $\alpha = 0,4$ ) y fuertes ( $\alpha = 0,6$ ) respectivamente.

## 4. Simulación SIR y SIRS con datos reales

Para comprobar cómo se ajustan los modelos SIR y SIRS a la realidad, realizamos una simulación para datos de la epidemia de ébola de 2014, en la que los países más afectados fueron Liberia, Sierra Leona y Guinea. La elección de esta epidemia no es trivial, pues al tratarse de países subdesarrollados, la imposición de medidas de contención se limitó a cuarentenas y, además, no había vacuna disponible para el ébola en ese momento, por lo que se trata de un caso de epidemia con pocas medidas de contención, por lo que podría ajustarse bien al funcionamiento de modelos sencillos como SIR y SIRS.

Usaremos los parámetros  $\beta = 0,0003589$ ,  $\nu = 0,1$  y  $\gamma = 0,0179$ , extraídos de [9]. Los datos reales fueron extraídos de [10]. En la **Figura 12** (cuaderno 5) se muestran estos datos frente a la simulación realizada con los dos de modelos más simples (SIR y SIRS), que no tienen en cuenta medidas de contención ni dinámicas vitales.



**Figura 12:** Comparación de la evolución de la epidemia de ébola en África Occidental según los modelos SIR y SIRS y evolución real.

Podemos observar que el modelo no se ajusta en absoluto a la evolución real de la epidemia. Esto se puede atribuir al hecho de no incluir dinámicas vitales, pues en nuestros modelos suponemos que la población se mantiene siempre constante, lo que nos deja con un modelo demasiado simplista, tanto en el caso del SIR como en el SIRS. Además, los datos que se han podido recuperar de la epidemia de ébola solo contienen los casos de infectados entre 2014 y 2016, cuando aún la epidemia no había sido controlada. Con todo, los modelos no proveen de una predicción realista de la evolución en este caso, dando la pandemia por terminada mucho antes de lo ocurrido.

Para una modelización precisa se necesitarán parámetros ajustados al contexto epidemiológico de la zona, además de las circunstancias demográficas y socioeconómicas de la zona. Otra posibilidad es explorar modelos más complejos como el SEIR, SEIQR, SEIQRV... [11] Así, al incluir la variación demográfica, el efecto epidemiológico de las medidas de contención, etc. en un modelo más adecuado para el tipo de epidemia, se obtendrá un resultado mucho más ajustado a la realidad.

## 5. Modelos epidemiológicos avanzados

Con la pandemia global del SARS-CoV-2, nuevos enfoques fueron desarrollados e implementados con el fin de obtener una predicción más precisa de la evolución epidemiológica. Se tuvieron en cuenta medidas de contención impuestas por los gobiernos como la cuarentena en modelos como el SEIQR, que incluía también un compartimento para la población expuesta y otro para la población en cuarentena.

Otro de los modelos aplicados fue el SIDARTHE, ideado en primer lugar por Gumel y compañía para la epidemia del SARS en 2003 [12]. A diferencia de los modelos anteriores, este hace una compartimentalización más minuciosa en los siguientes grupos: Susceptible, Infected (asintomáticos), Diagnosed (asintomáticos diagnosticados), Recognized (sintomáticos diagnosticados), Ailing (enfermos sin diagnosticar), Threatened (enfermos con infección aguda y grave), Healed (recuperados), y Extinct (muertos). Este modelo se basa en ecuaciones diferenciales fraccionarias y se ha visto que ofrece una modelización más precisa en comparación con los modelos clásicos SIR y SIRS.

En 2022, Chadi y Mousannif propusieron el uso de este modelo junto con aprendizaje reforzado como herramienta para predecir de forma precisa la evolución de la pandemia [13]. A diferencia de los modelos clásicos SIR y SIRS, SIDARTHE al tener una mayor compartimentalización, ofrece una visión más completa de la situación.

## 6. Conclusiones

La complejidad del análisis de los modelos epidemiológicos aumenta considerablemente a medida que aumentan el número de componentes en el sistema con la intención de ajustarse a la realidad. En cuanto a los modelos clásicos, como el SIR y el SIRS, observamos poca coincidencia con datos reales, aunque logran contener algunos conceptos básicos de las epidemias. En concreto, en el modelo SIRS observamos el fenómeno de la endemia, que se da o no en función del valor del número básico de reproducción ( $R_0$ ). Por su parte, el modelo SEIR, más reciente, puede ser modificado para ajustarse a más fenómenos de una epidemia actual, como la incubación de la enfermedad, las medidas gubernamentales de contención o la sensación de riesgo general entre la población. En cuanto a modelos epidemiológicos más avanzados, como el SIDARTHE, logran ajustarse en mayor medida a la realidad a coste de incrementar notablemente su complejidad y su profundidad matemática.

## 7. Bibliografía

- (1) Anderson, R. M. y May, R. M. Population biology of infectious diseases: Part I. *Nature* **1979**, *280*, 361-367.
- (2) Kermack, W. O. y McKendrick, A. G. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* **1927**, *115*, 700-721.
- (3) Delamater, P. L.; Street, E. J.; Leslie, T. F.; Yang, Y. T. y Jacobsen, K. H. Complexity of the basic reproduction number ( $R_0$ ). *Emerging infectious diseases* **2019**, *25*, 1.
- (4) Perko, L., *Differential equations and dynamical systems*; Springer Science & Business Media: 2013; vol. 7.
- (5) Chalub, F. A. C. C. y Souza, M. O. The SIR epidemic model from a PDE point of view. *Math. Comput. Model.* **2009**, *53*, 1568-1574.
- (6) Bailey, N. T. et al., *The mathematical theory of infectious diseases and its applications*; Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE.: 1975.
- (7) Kadkhoda, K. Herd immunity to COVID-19: alluring and elusive, 2021.
- (8) Gutiérrez, J. M. y Varona, J. L. Análisis de la posible evolución de la epidemia de coronavirus COVID-19 por medio de un modelo SEIR. *Departamento de Matemáticas y Computación Universidad de La Rioja* **2020**.
- (9) Agbomola, J. O. y Loyinmi, A. C. Optimal Control of Ebola Transmission Dynamics with Interventions. **2016**, *19(1)*, 1-19.
- (10) Center for Disease Control and Prevention (CDC). Case Counts Accessed: 2022-05-26, <https://www.cdc.gov/vhf/ebola/history/2014-2016-outbreak/case-counts.html>.
- (11) Li, M.-T.; Sun, G.-Q.; Zhang, J.; Zhao, Y.; Pei, X.; Li, L.; Wang, Y.; Zhang, W.-Y.; Zhang, Z.-K. y Jin, Z. Analysis of COVID-19 transmission in Shanxi Province with discrete time imported cases. *Mathematical Biosciences and Engineering* **2020**, *17*, 3710-3720.
- (12) Gumel, A. B.; Ruan, S.; Day, T.; Watmough, J.; Brauer, F.; van den Driessche, P.; Gabrielson, D.; Bowman, C.; Alexander, M. E.; Ardal, S.; Wu, J. y Sahai, B. M. Modelling strategies for controlling SARS outbreaks. *Proceedings of the Royal Society of London. Series B: Biological Sciences* **2004**, *271*, 2223-2232.
- (13) Chadi, M.-A. y Mousannif, H. A reinforcement learning based decision support tool for epidemic control: validation study for COVID-19. *Applied Artificial Intelligence* **2022**, *36*, 2031821.



## 8. Anexo

En un sistema cerrado, la población total se mantiene constante:

$$\begin{aligned}
 N &= S + I + R \\
 \frac{dN}{dt} &= \frac{d(S + I + R)}{dt} = \frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} \\
 \frac{dN}{dt} &= (-\beta IS) + (\beta IS - \nu I) + (\nu I) = 0 \\
 \frac{dN}{dt} &= 0 \rightarrow N \text{ no varía}
 \end{aligned} \tag{16}$$

Cálculo de los puntos críticos del modelo SIR:

$$\begin{cases} S'(t) = -\beta SI \\ I'(t) = \beta SI - \nu I \\ R'(t) = \nu I \end{cases}$$

$$\beta SI - \nu I = 0 \rightarrow I(\beta S - \nu) = 0$$

Primer caso, si  $I=0$ :

$$\begin{aligned}
 N &= S + I + R \\
 N &= S + R \\
 S &= N - R \\
 P_{c_1} &= (S, 0, N - S)
 \end{aligned}$$

Encontramos, por tanto, infinitos puntos críticos.

Podemos definir el espacio:

$$P\{(S, 0, N - S) \mid S, N \in \mathbb{R}^+ \mid S \leq N\} \tag{17}$$

Segundo caso, si  $S\beta - \nu = 0$ :

$$\begin{aligned}
 S\beta - \nu &= 0 \rightarrow S = \frac{\nu}{\beta} \\
 R' = \nu I &= 0 \rightarrow I = 0 \\
 N &= S + I + R \\
 R = N - S &\rightarrow R = N - \frac{\nu}{\beta} \\
 P_{c_2} &= \left( \frac{\nu}{\beta}, 0, N - \frac{\nu}{\beta} \right)
 \end{aligned}$$

Este punto crítico estaría incluido en el espacio  $P$ :

$$P_{c_2} \in P$$

Predicción de la tendencia del sistema SIR:

$$\begin{aligned}
& \text{PVI: } (S_i, I_i, R_i); \nu; \beta \\
& (S_i, I_i, R_i) \rightarrow (S_f, I_f, R_f) \\
& S_f = m(\nu, \beta) \cdot S_i \\
& I_f = 0 \\
& R_f = R_i + S_i + I_i - S_f
\end{aligned} \tag{18}$$

Cálculo de los puntos críticos del modelo SIRS:

$$\begin{cases} S'(t) = -\beta SI + \gamma R \\ I'(t) = \beta SI - \nu I \\ R'(t) = \nu I - \gamma R \end{cases}$$

$$\beta SI - \nu I = 0 \quad \rightarrow \quad I(\beta S - \nu) = 0$$

Primer caso, si  $I = 0$ :

$$\begin{aligned}
R' = \nu I - \gamma R = 0 & \rightarrow R = 0 \\
S = N - I - R = N
\end{aligned}$$

$$Pc_1 = (N, 0, 0)$$

Segundo caso, si  $\beta S - \nu = 0$ :

$$\begin{aligned}
\beta S - \nu = 0 & \rightarrow S = \frac{\nu}{\beta} \\
S'(t) = -\beta \frac{\nu}{\beta} I + \gamma R = -\nu I + \gamma R = 0 \\
-\nu I + \gamma \left( N - \frac{\nu}{\beta} - I \right) = -\nu I + \gamma N - \frac{\gamma \nu}{\beta} - \gamma I = 0 \\
I = \frac{\gamma(\beta N - \nu)}{\beta(\nu + \gamma)} \\
S'(t) = -\nu I + \gamma R = 0 \rightarrow R = \frac{\nu I}{\gamma} = \frac{\nu}{\gamma} \left( \frac{\gamma(\beta N - \nu)}{\beta(\nu + \gamma)} \right) \\
R = \frac{\nu \beta N - \nu^2}{\beta(\nu + \gamma)}
\end{aligned} \tag{19}$$

$$Pc_2 = \left( \frac{\nu}{\beta}, \frac{\gamma(\beta N - \nu)}{\beta(\nu + \gamma)}, \frac{\nu \beta N - \nu^2}{\beta(\nu + \gamma)} \right)$$

Intento de dejar el segundo punto crítico (puesto que el primero solo depende de  $N$ ) en función de  $R_0$  y no de  $\nu$  y  $\beta$ :

$$P_{e2} = \left( \frac{\nu}{\beta}, \frac{\gamma(N\beta - \nu)}{\beta(\gamma + \nu)}, \frac{\nu(N\beta - \nu)}{\beta(\gamma + \nu)} \right)$$

Queremos:  $P_{e2}(N, \nu, \beta, \gamma) \rightarrow P_{e2}(N, \gamma, R_0)$

$$R_0 = \frac{N\beta}{\nu} \quad \beta = \frac{R_0\nu}{N} \quad \nu = \frac{N\beta}{R_0}$$

$$P_{e2,S} = \frac{\nu}{\beta} = \frac{N}{R_0}$$

Para la I, primero intentamos con  $\beta$  :

$$P_{e2,I} = \frac{\gamma(N\frac{R_0\nu}{N} - \nu)}{\frac{R_0\nu}{N}(\gamma + \nu)} = \frac{\gamma N\nu(R_0 - 1)}{R_0\nu(\gamma + \nu)} = \frac{\gamma N(R_0 - 1)}{R_0(\gamma + \nu)} \quad (20)$$

Ahora intentamos con  $\nu$  :

$$P_{e2,I} = \frac{\gamma(N\beta - \frac{N\beta}{R_0})}{\beta(\gamma + \frac{N\beta}{R_0})} = \frac{\gamma N\beta(1 - \frac{1}{R_0})}{\beta(\gamma + \frac{N\beta}{R_0})} = \frac{\gamma N(1 - \frac{1}{R_0})}{\gamma + \frac{N\beta}{R_0}}$$

No podemos, en ambos casos nos queda  $\nu$  o  $\beta$

# MODELO SIR

El modelo en un principio está regido por un sistema de ecuaciones diferenciales ordinarias (EDO) de 3 ecuaciones tal que:

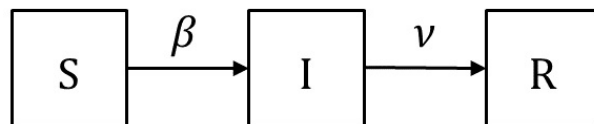
$$\begin{cases} \frac{dS}{dt} = -\beta SI \\ \frac{dI}{dt} = \beta SI - \nu I \\ \frac{dR}{dt} = \nu I \end{cases}$$

Donde: -> S: susceptibles. Pueden contagiarse. -> I: infectados. Están contagiados y pueden contagiar a otros -> R: recuperados. Gente curada o muerta, ya no puede contraer la enfermedad. ->  $\beta$ : tasa de infección ->  $\nu$ : tasa de curación o defunción

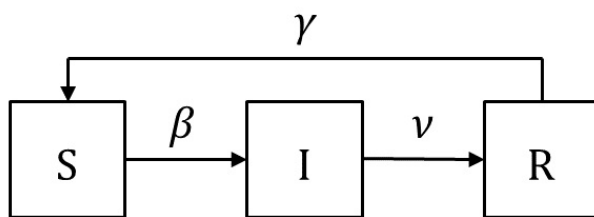
Al ser un modelo compartimental, podemos representarlo como un diagrama de cajas:

```
In [ ]: from IPython.display import Image
Image(filename='figura_modelos_compartimentales.png')
```

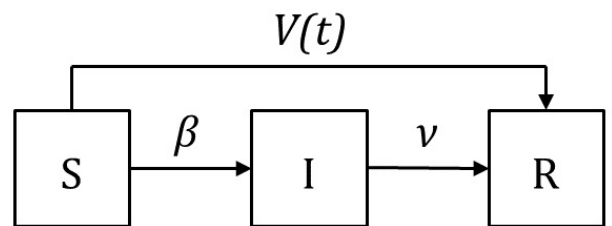
Out[ ]:



(a)



(b)



(c)

Nuestro modelo SIR sería el (a), y podemos calcular sus puntos críticos (de equilibrio) de la siguiente forma:

$$\begin{cases} S'(t) = -\beta SI \\ I'(t) = \beta SI - \nu I \\ R'(t) = \nu I \end{cases}$$

$$\beta SI - \nu I = 0 \rightarrow I(\beta S - \nu) = 0$$

Primer caso, si  $I=0$ :

$$N = S + I + R$$

$$N = S + R$$

$$S = N - R$$

$$Pc_1 = (S, 0, N - S)$$

Encontramos, por tanto, infinitos puntos críticos.

Podemos definir el espacio:

$$P\{(S, 0, N - S) \mid S, N \in \mathbb{R}^+ \mid S \leq N\}$$

Segundo caso, si  $\beta S - \nu = 0$ :

$$\beta S - \nu = 0 \rightarrow S = \frac{\nu}{\beta}$$

$$R' = \nu I = 0 \rightarrow I = 0$$

$$N = S + I + R$$

$$R = N - S \rightarrow R = N - \frac{\nu}{\beta}$$

$$Pc_\infty = \left( \frac{\nu}{\beta}, 0, N - \frac{\nu}{\beta} \right)$$

Este punto crítico estaría incluido en el espacio  $P$ :

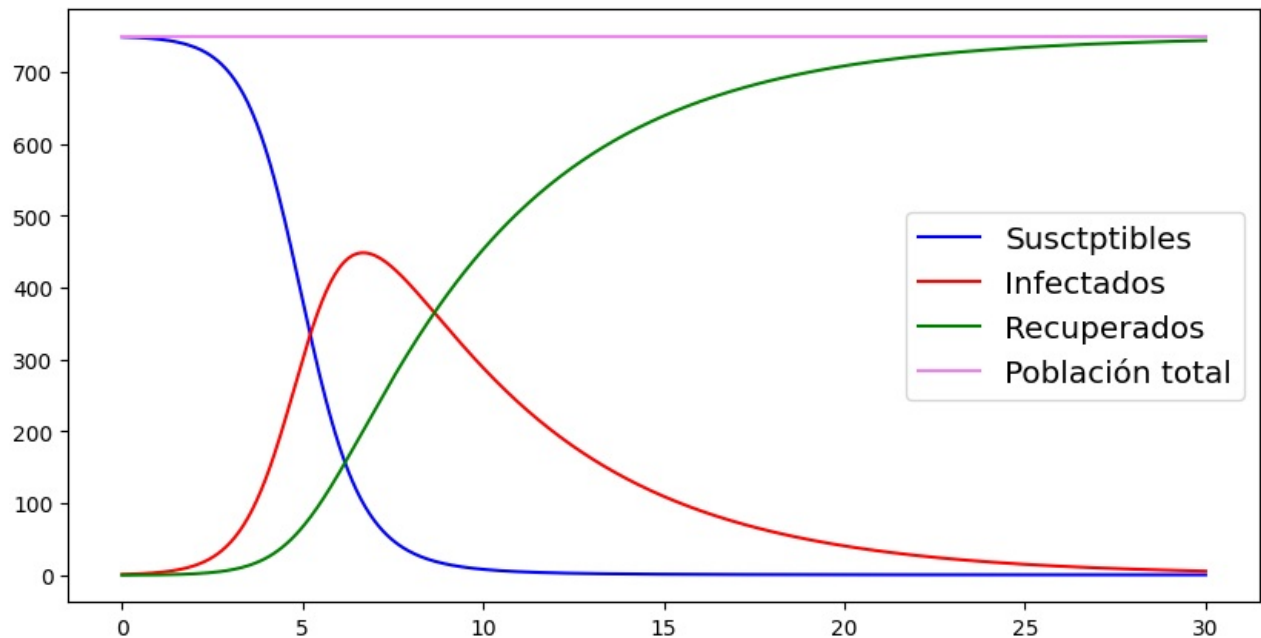
$$Pc_2 \in P$$

Importamos los paquetes necesarios:

```
In [1]: import matplotlib.pyplot as plt
import scipy.integrate as scint
import sympy as sp
import numpy as np
sp.init_printing()
```

Primero se reperestrna gráficamente la evolución de las poblaciones con respecto al tiempo

```
In [2]: beta=0.002;nu=0.2
def F(P, t):
    return [-beta*P[0]*P[1], beta*P[0]*P[1]-nu*P[1], nu*P[1]]
tiempo = np.linspace(0, 30,500)
P0 = [749, 1, 0]
Ps = scint.odeint(F, P0, tiempo)
fig=plt.figure(figsize=(10,5))
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscpttibles")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados")
plt.plot(tiempo,Ps[:,0] + Ps[:,1] + Ps[:,2],color="violet",label="Población total")
plt.legend(fontsize='x-large')
# plt.title(r'Modelo SIR: $P_0=(749,1,0)$ $\beta=0.002$ $\nu=0.2$', fontsize=15)
plt.savefig('gráfica_simple_SIR.png')
plt.show()
```



Los susceptibles se contagian y los infectados se curan. El resultado es un pico de infección que se resuelve finalmente, de modo que la epidemia termina.

Veamos cómo influyen los parámetros en la evolución de la epidemia:

```
In [3]: fig=plt.figure("figuraprueba",figsize=(12,8))
tiempo = np.linspace(0, 30,500)

plt.subplot(221)
beta=0.002; nu=0.2
Ps = scint.odeint(F, P0, tiempo)
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscpttibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.legend(fontsize='large')
plt.title(r'(a) $\beta = 0.002 \quad \nu = 0.2$', fontsize=15)
plt.ylabel('hospedadores (S, I, R)',fontsize=13)

plt.subplot(222)
beta=0.002; nu=0.6
Ps = scint.odeint(F, P0, tiempo)
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscpttibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.title(r'(b) $\beta = 0.002 \quad \nu = 0.5$', fontsize=15)
plt.legend(fontsize='large')
print(Ps[-1,0])
```

```

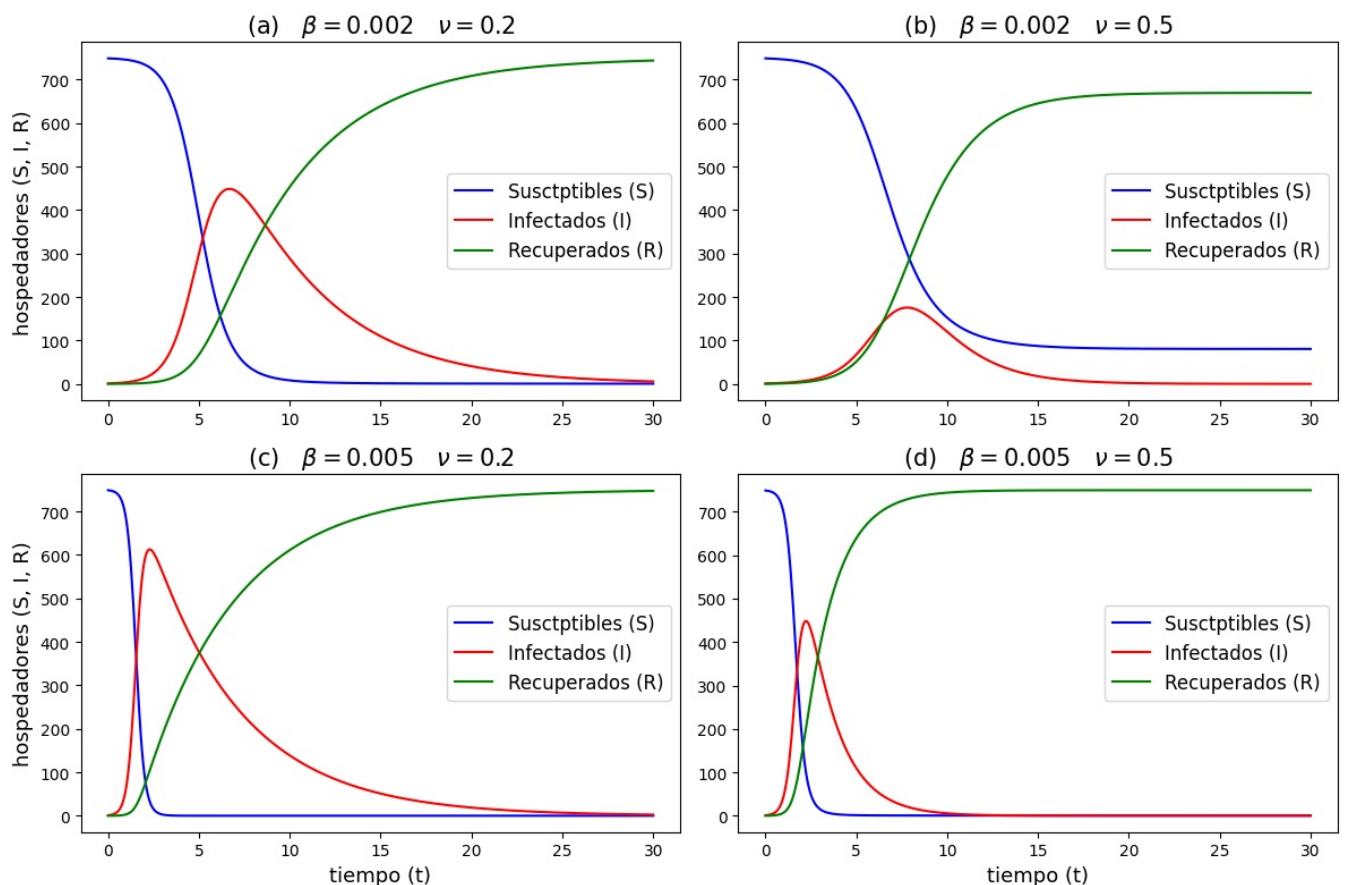
plt.subplot(223)
beta=0.006; nu=0.2
Ps = scint.odeint(F, P0, tiempo)
plt.plot(tiempo, Ps[:,0], color="blue",label="Susceptibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.legend(fontsize='large')
plt.title(r'(c) $\beta = 0.005 \backslash \text{quad} \backslash \nu = 0.2$', fontsize=15)
plt.xlabel('tiempo (t)',fontsize=13)
plt.ylabel('hospedadores (S, I, R)',fontsize=13)

plt.subplot(224)
beta=0.006; nu=0.6
Ps = scint.odeint(F, P0, tiempo)
plt.plot(tiempo, Ps[:,0], color="blue",label="Susceptibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.legend(fontsize='large')
plt.title(r'(d) $\beta = 0.005 \backslash \text{quad} \backslash \nu = 0.5$', fontsize=15)
plt.xlabel('tiempo (t)',fontsize=13)

fig.tight_layout()
plt.savefig('efecto_nubeta_SIR.png')
plt.show()

```

80.37882945133103



Aparentemente, se vé cómo, al aumentar  $\beta$  (manteniendo  $\nu$  constante), los susceptibles caen más rápido y el pico de infectados sube más alto. La interpretación es sencilla, ya que si la tasa de infección es mayor, los infectados suben más rápido y los susceptibles caen más rápido, lo que genera un pico de infección más alto. Por otro lado, al aumentar  $\nu$  (manteniendo  $\beta$  constante), parece que el pico de infectados se hace más corto y estrecho. Esto podría deberse a que, ya que al curación es más rápida, el número máximo de infectados es menor y la infección se resuelve antes. Es como si "tiramos" de S hacia R.

Parece que el destino de este sistema es que todos los susceptibles se contagian y todos los infectados se curan. Es decir, podría pensarse como una cascada de S a I y de I a R. Un flujo inescrutible que lleva al sistema al punto  $(0,0,N)$ , que correspondería a uno de los infinitos puntos críticos del sistema. Sin embargo, en la gráfica superior derecha vemos que la población final de susceptibles no es 0, sino que queda una cierta población de susceptibles que ha "sobrevivido" a la epidemia. ¿qué diferencia a este caso del resto? ¿existe siquiera alguna diferencia realmente, o simplemente en el resto de casos la población de supervivientes es muy pequeña? ¿Cómo se relaciona esto con la estabilidad de los puntos críticos?

Podemos obtener algo de información observando las poblaciones finales en el caso del principio ( $P_0 = (749, 1, 0)$ ;  $\beta = 0.02$  y  $\nu = 0.2$ ) tras mucho tiempo:

```

In [7]: beta=0.002;nu=0.2;P0 = [749, 1, 0]
tiempo = np.linspace(0, 4000,20000)
Ps = scint.odeint(F, P0, tiempo)
#El sistema ha tendido hacia:

```

```
print(Ps[-1,0],Ps[-1,1],Ps[-1,2])
```

```
0.4159870244503032 -6.398339881625687e-19 749.5840129755502
```

Vemos que, a diferencia de los infectados, la población de susceptibles no tiende a 0, sino que en este caso solamente es muy pequeña en comparación con la población total.

En cualquier caso, observamos que, del conjunto de total de puntos críticos, sólo uno de ellos es asintóticamente estable para un determinado PVI. Cabría preguntarse... ¿Podemos predecir estas poblaciones finales a partir de un PVI concreto?

## Estudio cualitativo del sistema

```
In [8]: s, i, r = sp.symbols("S I R")
b, n, v = sp.symbols("beta N nu", positive=True)
fsi = b*s*i
gsi = b*s*i - v*i
r = n - s - i
fsi, gsi
```

```
Out[8]: (-Iβ, Iβ - Iv)
```

```
In [9]: p_cr = sp.solve([fsi, gsi], [s, i], simplify=True)
p_cr
```

```
Out[9]: [(S, 0)]
```

Vemos que, en efecto, el sistema presenta un punto crítico  $(S, 0, N - S)$  que, en realidad, representa un conjunto infinito de puntos críticos donde  $0 < S \leq N$  (significado biológico).

```
In [10]: par_matrix = sp.Matrix([[fsi.diff(s), fsi.diff(i)], [gsi.diff(s), gsi.diff(i)]])
par_matrix
```

```
Out[10]: 
$$\begin{bmatrix} -I\beta & -S\beta \\ I\beta & S\beta - v \end{bmatrix}$$

```

### Análisis para el primer y único punto crítico:

```
In [11]: C1 = par_matrix.subs({s:p_cr[0][0], i:p_cr[0][1]})
C1
```

```
Out[11]: 
$$\begin{bmatrix} 0 & -S\beta \\ 0 & S\beta - v \end{bmatrix}$$

```

```
In [12]: Clauto = C1.eigenvals()
Clautovect = C1.eigenvects()
Clauto, Clautovect
```

```
Out[12]: 
$$\left( \{0:1, S\beta - v:1\}, \left[ \left( 0, 1, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right), \left( S\beta - v, 1, \begin{bmatrix} -\frac{S\beta}{S\beta - v} \\ 1 \end{bmatrix} \right) \right] \right)$$

```

```
In [13]: trazaC1 = sp.simplify(sp.trace(C1))
detC1 = sp.simplify(sp.det(C1))
trazaC1, detC1
```

```
Out[13]: (Sβ - v, 0)
```

Condición de estabilidad asintótica del sistema lineal:  $\text{tr}(A) < 0$  ;  $\text{det}(A) > 0$

Esto es:

$$\begin{cases} S\beta - v < 0 \\ 0 > 0 \end{cases}$$

Vemos que el determinante de la matriz jacobiana evaluada en cualquier punto del tipo  $(S, 0, N - S)$  presenta un determinante nulo, por lo que no podemos conocer, a priori, la estabilidad. Tampoco podemos saber, a priori, cómo influye la traza en la estabilidad del sistema.

Es decir, no sabemos si el parámetro  $\frac{S\beta}{v}$  influye en algo, aunque, si lo hiciera no sabemos qué  $S$  sería, si la del PVI que tiende al punto o la del punto de equilibrio en sí (tendría más sentido que fuera la del punto al que se tiende, ya que la  $S$  inicial es algo bastante abstracto, pues la estabilidad de un punto no depende de en qué punto del pasado empezamos a simular el sistema. Es decir, el sistema no conoce "un principio" como tal).

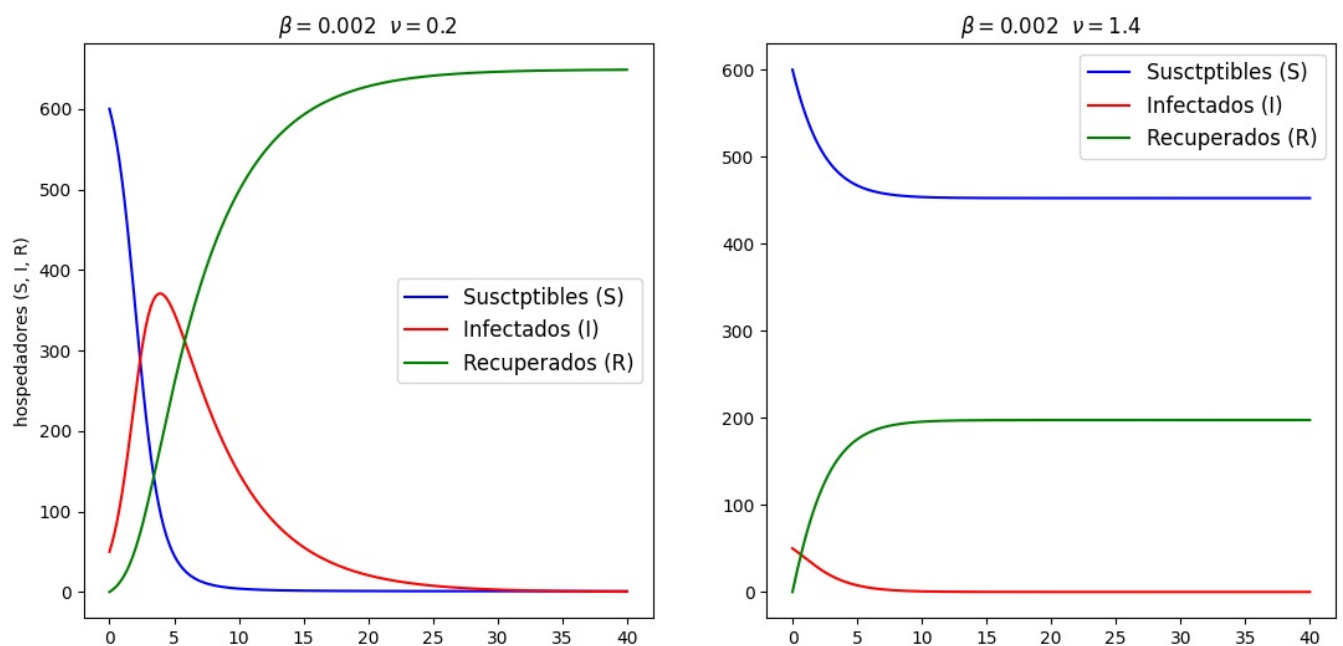
Fracción de supervivientes...

Vamos a ver qué diferencia hay entre los casos con  $\nu$  alta y baja.

```
In [14]: fig=plt.figure("figuraprueba",figsize=(13,6))
P0=[600,50,0]
tiempo=np.linspace(0,40,200)
plt.subplot(121, title = r'$\beta = 0.002$  $\nu = 0.2$')
beta=0.002; nu=0.2
Ps = scint.odeint(F, P0, tiempo)
final1=(Ps[-1,0],Ps[-1,1],Ps[-1,2])
sigma1=(Ps[-1,0]*beta/nu)
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscptibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.legend(fontsize='large')
plt.ylabel('hospedadores (S, I, R)')

plt.subplot(122, title = r'$\beta = 0.002$  $\nu = 1.4$')
beta=0.002; nu=1.4
Ps = scint.odeint(F, P0, tiempo)
final2=(Ps[-1,0],Ps[-1,1],Ps[-1,2])
sigma2=(Ps[-1,0]*beta/nu)
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscptibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.legend(fontsize='large')
```

Out[14]: <matplotlib.legend.Legend at 0x7f29838536d0>



Vale, vamos a ver las poblaciones finales:

```
In [ ]: print(final1)
print(final2)
print(sigma1)
print(sigma2)

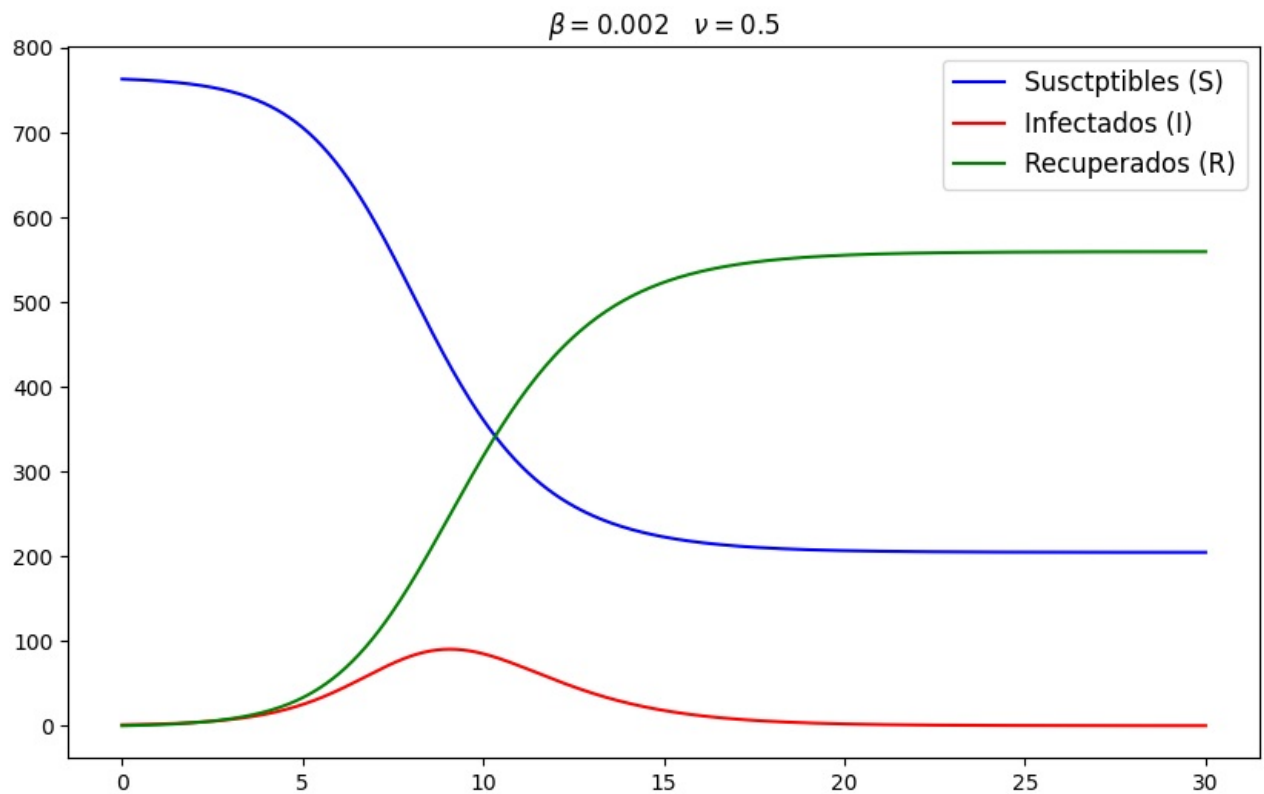
(0.9139516944900775, 0.3953231001802054, 648.6907252053302)
(452.4987633249875, 2.416525329193376e-07, 197.5012364333601)
0.009139516944900774
0.6464268047499822
```

Pues esto no aporta mucho. La diferencia es cuantitativa más que cualitativa. Simplemente vemos que al aumentar el ratio  $\nu/\beta$  aumenta la población de supervivientes (que recordemos que es el valor de S al que tiende el sistema).

```
In [ ]: fig=plt.figure("figuraprueba",figsize=(10,6))
tiempo = np.linspace(0, 30,500)
P0 = [763, 1, 0]
beta=0.002; nu=0.85
Ps = scint.odeint(F, P0, tiempo)
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscptibles (S)")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados (I)")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados (R)")
plt.legend(fontsize='large')
plt.title( r'$\beta = 0.002$ \quad \nu = 0.5$')
```

Out[ ]: Text(0.5, 1.0, '\$\beta = 0.002\$ \quad \nu = 0.5\$')





¿A qué valores está tendiendo el sistema?

```
In [ ]: print(Ps[-1,0],Ps[-1,1],Ps[-1,2])
204.61095374736274 0.026301965746725647 559.3627442868902
```

¿Coincide esto con el punto de equilibrio  $\left(\frac{\nu}{\beta}, 0, N - \frac{\nu}{\beta}\right)$ ?

```
In [ ]: print(nu/beta, 0, (Ps[-1,0]+Ps[-1,1]+Ps[-1,2]) - (nu/beta))
425.0 0 338.9999999999998
```

No coincide. ¿Cómo puedo predecir este punto asintóticamente estable a partir de los parámetros y condiciones iniciales?

Pues bueno, voy a hacer un código con el que voy variando  $\nu$  y comparo el  $S_f$  con  $\nu/\beta$ ... a ver qué pasa

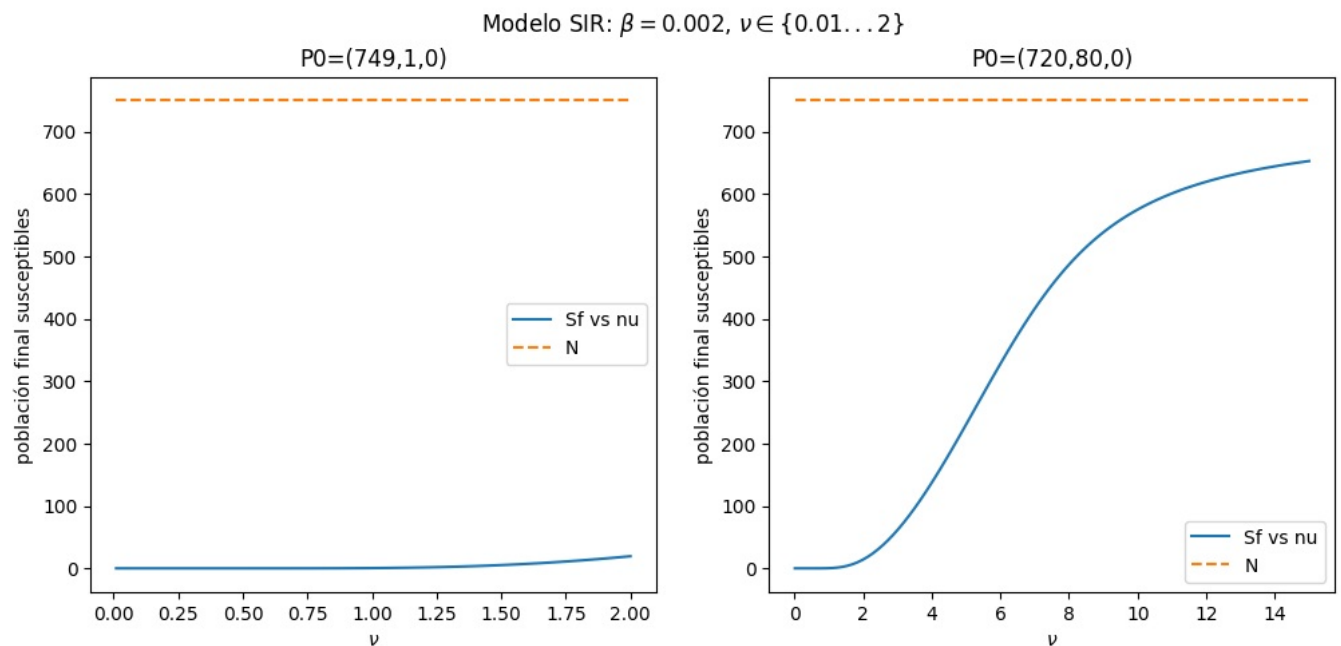
```
In [16]: gradsize=100
tiempo = np.linspace(0,500,1000)
N=750; solo_N=np.ones(gradsize)*N
fig=plt.figure(figsize=(12,5))

plt.subplot(121, title='P0=(749,1,0)')
grad_nu=np.linspace(0.01,2,gradsize)
P0 = [749, 1, 0]
array_Sf=list()
array_nu_bet=list()
array_bet_nu=list()
array_S_bet_nu=list()
array_R0=list()
for nu in grad_nu:
    Ps=Ps = scint.odeint(F, P0, tiempo)
    array_Sf.append(Ps[-1,0])
    array_nu_bet.append(nu/beta)
    array_bet_nu.append(beta/nu)
    array_S_bet_nu.append(Ps[-1,0]*beta/nu)
    array_R0.append(N*beta/nu)
plt.plot(grad_nu,array_Sf,label='Sf vs nu')
# plt.plot(array_nu_bet,array_Sf,label='Sf vs nu/beta')
# plt.plot(array_bet_nu,array_Sf,label='Sf vs beta/nu')
# plt.plot(array_R0,array_Sf,label='Sf vs R0')
# plt.plot(grad_nu,array_nu_bet,label='nu/beta')
# plt.plot(grad_nu,array_bet_nu,label='beta/nu')
# plt.plot(grad_nu,array_S_bet_nu,label='S*beta/nu')
# plt.plot(grad_nu,array_R0,label='R0')
plt.plot(grad_nu,solo_N,linestyle='--',label='N')
plt.legend()
plt.xlabel(r'$\nu$')
```

```
plt.ylabel('población final susceptibles')

plt.subplot(122, title='P0=(720,80,0)')
grad_nu=np.linspace(0.01,15,gradsize)
P0 = [720, 80, 0];
array_Sf=list()
array_nu_bet=list()
array_bet_nu=list()
array_S_bet_nu=list()
array_R0=list()
for nu in grad_nu:
    Ps=Ps = scint.odeint(F, P0, tiempo)
    array_Sf.append(Ps[-1,0])
    array_nu_bet.append(nu/beta)
    array_bet_nu.append(beta/nu)
    array_S_bet_nu.append(Ps[-1,0]*beta/nu)
    array_R0.append(N*beta/nu)
plt.plot(grad_nu,array_Sf,label='Sf vs nu')
# plt.plot(array_nu_bet,array_Sf,label='Sf vs nu/beta')
# plt.plot(array_bet_nu,array_Sf,label='Sf vs beta/nu')
# plt.plot(array_R0,array_Sf,label='Sf vs R0')
# plt.plot(grad_nu,array_nu_bet,label='nu/beta')
# plt.plot(grad_nu,array_bet_nu,label='beta/nu')
# plt.plot(grad_nu,array_S_bet_nu,label='S*beta/nu')
# plt.plot(grad_nu,array_R0,label='R0')
plt.plot(grad_nu,solo_N,linestyle='--',label='N')
plt.legend()
plt.xlabel(r'$\nu$')
plt.ylabel('población final susceptibles')

# plt.xlim(0,10)
fig.suptitle(r'Modelo SIR: $\beta=0.002$, $\nu\in\{0.01\dots2\}$')
plt.show()
```



Estas gráficas nos dan mucha información... Se obtienen resultados similares al graficar frente a  $\nu/\beta$

```
In [17]: gradsize=200
N=750; solo_N=np.ones(gradsize)*N
tiempo = np.linspace(0,500,1000)
fig=plt.figure(figsize=(12,5))

plt.subplot(121, title='P0=(749,1,0)')
grad_R0=np.linspace(0.01,7.5,gradsize)
tiempo = np.linspace(0,100,500)
P0 = [749, 1, 0]
array_Sf=list()
for R0 in grad_R0:
    nu=N*beta/R0
    Ps=Ps = scint.odeint(F, P0, tiempo)
    array_Sf.append(Ps[-1,0])
plt.plot(grad_R0,array_Sf,label='Sf vs R0')
plt.plot(grad_R0,solo_N,linestyle='--',label='N')
plt.axvline(x=1, color='r', linestyle='--')
plt.legend()
plt.xlabel(r'$R0$')
plt.ylabel('población final susceptibles')

plt.subplot(122, title='P0=(720,80,0)')
grad_R0=np.linspace(0.01,7.5,gradsize)
P0 = [720, 80, 0]
```

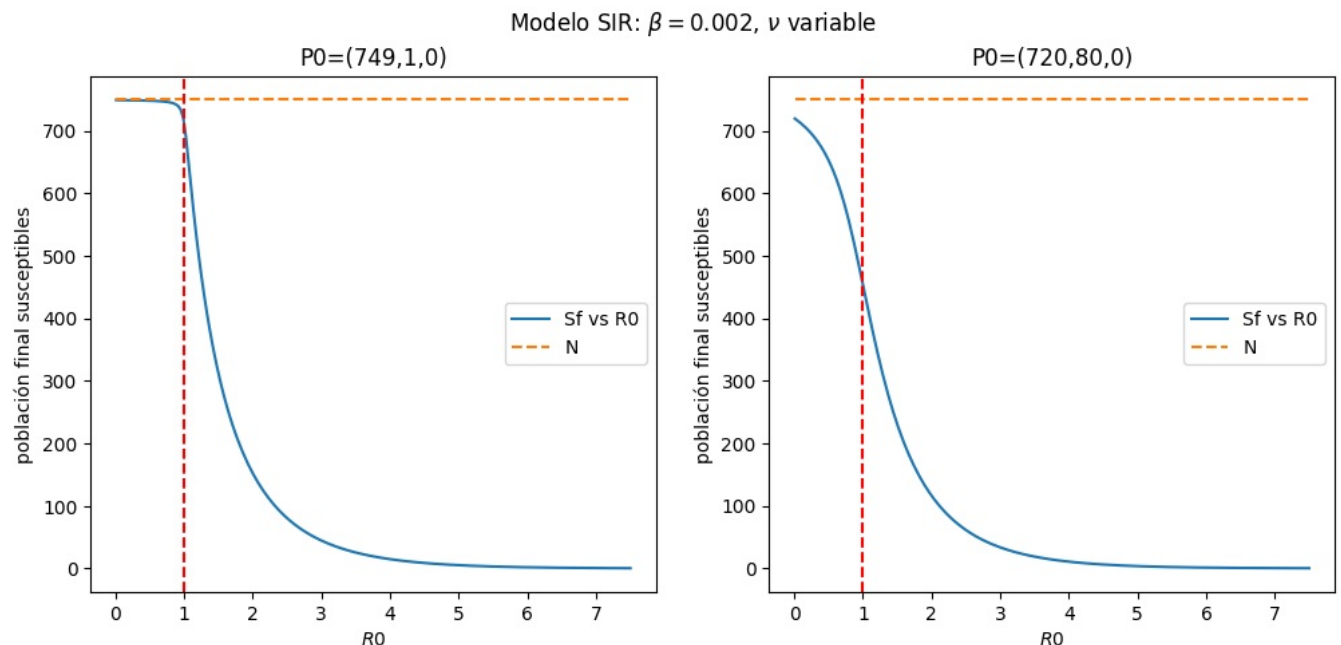
```

array_Sf=list()
for R0 in grad_R0:
    nu=N*beta/R0
    Ps=Ps = scint.odeint(F, P0, tiempo)
    array_Sf.append(Ps[-1,0])
plt.plot(grad_R0,array_Sf,label='Sf vs R0')
plt.plot(grad_R0,solo_N,linestyle='--',label='N')
plt.axvline(x=1, color='r', linestyle='--')
plt.legend()
plt.xlabel(r'$R_0$')
plt.ylabel('población final susceptibles')

fig.suptitle(r'Modelo SIR: $\beta=0.002$, $\nu$ variable')

# plt.xlim(0,10)
plt.show()

```



## Relación entre $S_i$ y $S_f$

También cabe preguntarse si la población de supervivientes depende del PVI, lo cual nos esperamos que sí dependa pero claramente no tenemos ni idea de cómo es, en concreto, esta relación. Consideramos dos casos: ( $\beta = 0.002$  y  $\nu = 0.2$ ) y ( $\beta = 0.002$  y  $\nu = 0.5$ ). En ambos casos trazamos la curva con múltiples condiciones iniciales, concretamente en el rango: ( $\{0...749\}$ ,  $\{750...1\}$ , 0).

```

In [18]: tiempo = np.linspace(0,500,1000)
N=750; gradsize=N; solo_N=np.ones(gradsize)*N
fig=plt.figure(figsize=(10,4))

plt.subplot(121, title=r'$\beta=0.002$ y $\nu=0.3$')
grad_S=range(750)
beta=0.002; nu=0.3
array_Sf=list()
array_SoN=list()
for S in grad_S:
    P0=[S,N-S,0]
    Ps = scint.odeint(F, P0, tiempo)
    array_Sf.append(Ps[-1,0])
    array_SoN.append(S/N)
plt.plot(grad_S,array_Sf,label=r'$S_f$ vs $S_i$')
plt.plot(grad_S,solo_N,linestyle='--',label='N={}'.format(N))
Sff=Ps[-1,0]
solo_Sff=np.ones(gradsize)*Sff
plt.plot(grad_S,solo_Sff,linestyle='--',label='Sf={}'.format(Sff))
m=Ps[-1,0]/grad_S[-1]
plt.plot(grad_S,grad_S*m,linestyle='--',color='red',label='relación lineal')
plt.legend()
plt.ylim(0,70)
plt.xlim(0,750)
plt.xlabel(r'$S_i$', fontsize=15)
plt.ylabel(r'$S_f$', fontsize=15)

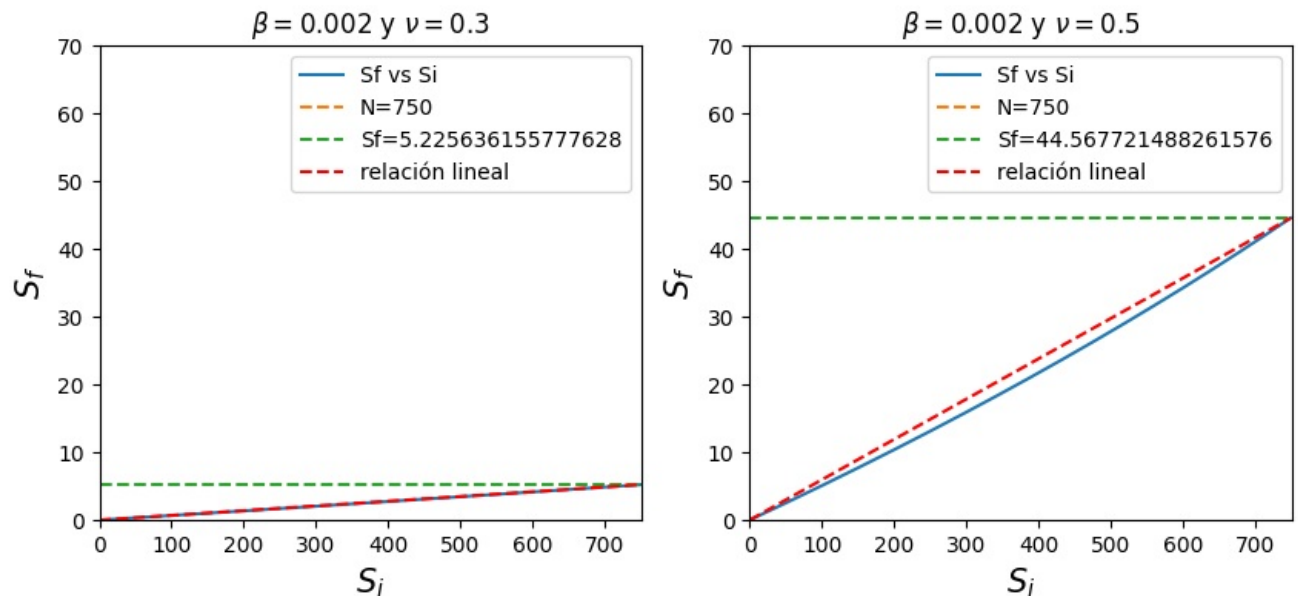
plt.subplot(122, title=r'$\beta=0.002$ y $\nu=0.5$')
grad_S=range(750)
beta=0.002; nu=0.5
array_Sf=list()
array_SoN=list()
for S in grad_S:
    P0=[S,N-S,0]

```

```

Ps = scint.odeint(F, P0, tiempo)
array_Sf.append(Ps[-1,0])
array_SoN.append(S/N)
plt.plot(grad_S,array_Sf,label=r'Sf vs Si')
plt.plot(grad_S,solo_N,linestyle='--',label='N={}'.format(N))
Sff=Ps[-1,0]
solo_Sff=np.ones(gradsize)*Sff
plt.plot(grad_S,solo_Sff,linestyle='--',label='Sf={}'.format(Sff))
m=Ps[-1,0]/grad_S[-1]
plt.plot(grad_S,grad_S*m,linestyle='--',color='red',label='relación lineal')
plt.legend()
plt.ylim(0,70)
plt.xlim(0,750)
plt.xlabel(r'$S_i$',fontsize=15)
plt.ylabel(r'$S_f$',fontsize=15)
fig.tight_layout
plt.savefig('graf_Sf_Si_SIR.png')
# fig.suptitle(r'Modelo SIR: $P_0$ in {(0,750,0)...(750,0,0)}$')
plt.show()

```



Parece que existe una relación lineal entre la fracción de susceptibles iniciales y la población de supervivientes (y, por tanto, la fracción de supervivientes... pero eso da igual). Además, vemos que los supervivientes aumentaron más de 10 veces al pasar de  $\nu = 0.2$  a  $\nu = 0.5$ . Entonces, ahora cabría preguntarse... qué relación hay entre esta pendiente ( $\Delta S_f / \Delta S_o$ ) y  $\nu$  y/o  $\beta$ .

Para ver esto vamos a ver cómo cambia la pendiente para el rango de  $\nu \{0.01 \dots 1.5\}$  y el rango  $\beta \{0.0001 \dots 0.01\}$

```

In [19]: # Esta celda tarda mucho en ejecutarse, según el valor de estas dos variables:
gradsize_nu=15; gradsize_beta=10 # Con estos valores se tarda unos 10 segundos
# Para el artículo se escogieron valores más altos para tener curvas más suaves
N=400; gradsize=N
tiempo=np.linspace(0,100,100)
grad_S=range(N)
fig=plt.figure(figsize=(10,4))

plt.subplot(121)
beta=0.002
grad_nu=np.linspace(0.01,1.5,gradsize_nu)
pendientes_m_nu=list()
for nu in grad_nu:
    array_Sf=list()
    array_SoN=list()
    for S in grad_S:
        P0=[S,N-S,0]
        Ps = scint.odeint(F, P0, tiempo)
        array_Sf.append(Ps[-1,0])
        array_SoN.append(S/N)
    m=(array_Sf[-1]-array_Sf[0])/(grad_S[-1]-grad_S[0])
    pendientes_m_nu.append(m)
plt.plot(grad_nu,pendientes_m_nu,color='blue')
plt.xlabel(r'$\nu$',fontsize=13)
plt.ylabel(r'$\Delta S_f / \Delta S_i$',fontsize=13)

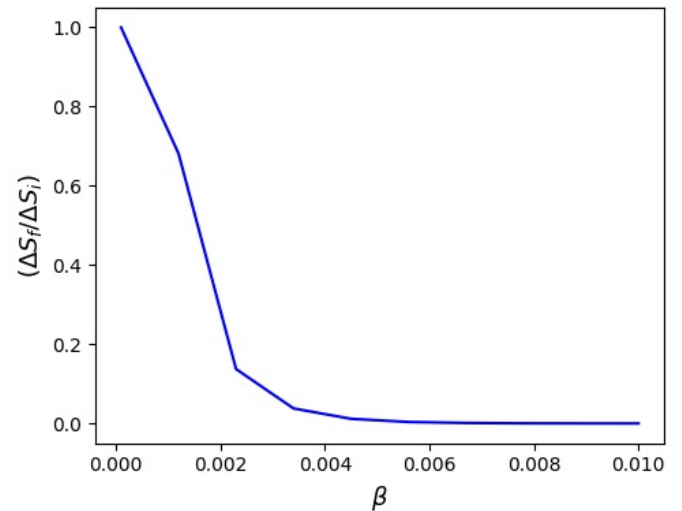
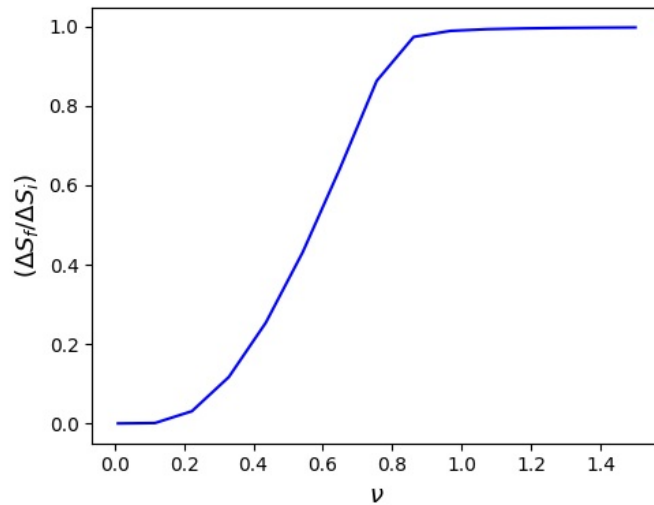
plt.subplot(122)
nu=0.4
grad_beta=np.linspace(0.0001,0.01,gradsize_beta)
pendientes_m_beta=list()
for beta in grad_beta:
    array_Sf=list()
    array_SoN=list()
    for S in grad_S:
        P0=[S,N-S,0]

```

```

Ps = scint.odeint(F, P0, tiempo)
array_Sf.append(Ps[-1,0])
array_SoN.append(S/N)
m=(array_Sf[-1]-array_Sf[0])/(grad_S[-1]-grad_S[0])
pendientes_m_beta.append(m)
plt.plot(grad_beta,pendientes_m_beta,color='blue')
plt.xlabel(r'$\beta$',fontsize=13)
plt.ylabel(r'$\Delta S_f / \Delta S_i$',fontsize=13)
fig.tight_layout()
plt.savefig('graf_SfSo_nubeta_SIR.png')
plt.show()

```



DOS COSAS: -> la relación entre  $(\Delta S_f / \Delta S_o)$  y  $\nu$  viene dada por una curva sigmoidea ->  $(\Delta S_f / \Delta S_o)$  más o menos decae exponencialmente respecto a  $\beta$ . Es decir, el  $-\log(\Delta S_f / \Delta S_o)$  aumenta linealmente con respecto a  $\beta$ . Esto ocurre en cierto rango. Es decir, tiene pinta de que la curva es sigmoidea (porque a valores de beta cercanos a 0, la pendiente se mantiene en 1). Sin embargo, al graficar  $-\log(\Delta S_f / \Delta S_o)$  respecto a  $\beta$  obtenemos:

```

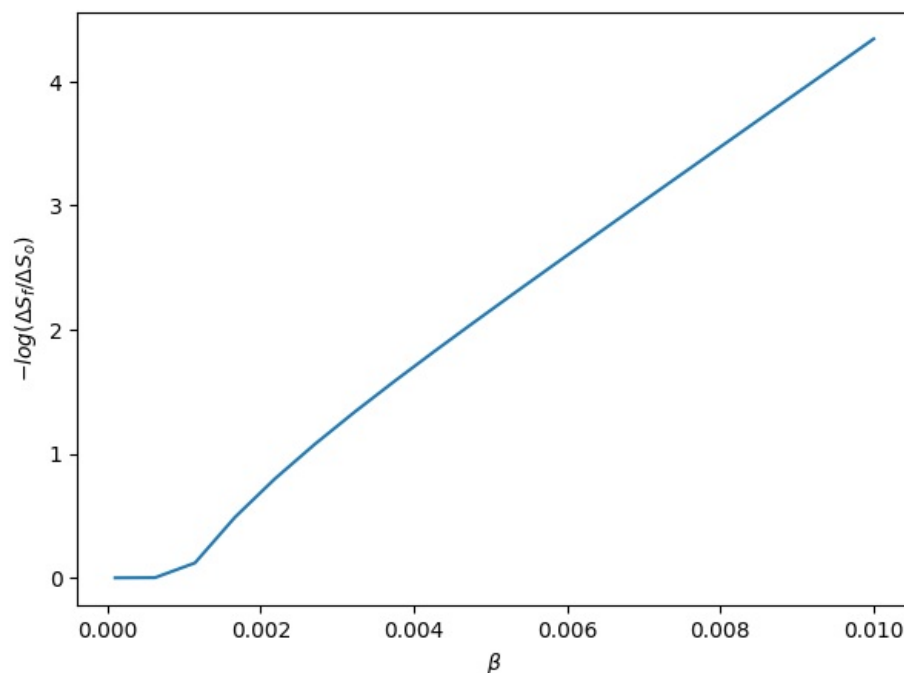
In [ ]: N=400; gradsize=N
tiempo=np.linspace(0,100,100)
grad_S=range(N)
fig=plt.figure(figsize=(7,5))
import math
nu=0.4
grad_beta=np.linspace(0.0001,0.01,20) #Con 20 puntos esto tarda unos 7 segundos
pendientes=list()
for beta in grad_beta:
    array_Sf=list()
    array_SoN=list()
    for S in grad_S:
        P0=[S,N-S,0]
        Ps = scint.odeint(F, P0, tiempo)
        array_Sf.append(Ps[-1,0])
        array_SoN.append(S/N)
        m=(array_Sf[-1]-array_Sf[0])/(grad_S[-1]-grad_S[0])
        pendientes.append(-math.log10(m))
plt.plot(grad_beta,pendientes)
plt.xlabel(r'$\beta$')
plt.ylabel(r'$-\log(\Delta S_f / \Delta S_o)$')

```

```

Out[ ]: Text(0, 0.5, '$-\log(\Delta S_f / \Delta S_o)$')

```



Se confirma un poco lo que decíamos. Parece lineal a partir de ciertos valores. Es decir, que aunque la relación real entre la pendiente y beta parezca ser más complicada, puede ser aproximada con una simple relación logarítmica (o exponencial, según se vea) en cierto rango de valores de beta, que, en realidad, puede que sean los que se den en la realidad. Un valor de  $\beta = 0.001$  significaría que de cada 1000 interacciones entre susceptibles e infecciosos, sólo 1 resulta en contagio, lo cual resulta poco realista.

## Estimación numérica de la tendencia con los parámetros del modelo

Ahora sólo faltaría ver cómo se relacionan numéricamente  $(\Delta S_f / \Delta S_0)$  con  $\nu$  y  $\beta$  para poder dilucidar una fórmula que nos permita obtener este ratio a partir de los parámetros del sistema. Esto sería suficiente para predecir hacia dónde evoluciona el sistema a partir de la condición inicial, puesto que sabemos que cualquier punto de equilibrio presenta  $I = 0$ , por lo que al saber  $(\Delta S_f / \Delta S_0)$  podemos saber  $S_f$  a partir de  $S_0$ , lo que nos permite obtener  $R_0 = N - S_0$ . Nótese que  $R_0$  da completamente igual, porque los recuperados no hacen absolutamente nada. Podemos ver un ejemplo de esto con una simulación numérica:

```
In [ ]: tiempo = np.linspace(0, 300,100)
P0 = [763, 1, 0]
beta=0.002; nu=0.85
Ps = scint.odeint(F, P0, tiempo)
print(Ps[-1,0],Ps[-1,1],Ps[-1,2])
P0 = [763, 1, 50]
beta=0.002; nu=0.85
Ps = scint.odeint(F, P0, tiempo)
print(Ps[-1,0],Ps[-1,1],Ps[-1,2])
```

204.58653760139458 -3.2163798603485316e-14 559.4134623986055  
204.5865417342635 -4.2071787698800844e-14 609.4134582657368

partiendo del PVI:  $P_0 = (763, 1, 0)$ ;  $\beta = 0.002$ ;  $\nu = 0.85$ ; vemos que tendemos hacia  $P_f \approx (204.59, 0, 559.41)$ . Si hubiéramos partido de  $P_0 = (763, 1, 50)$  acabaremos con los mismo susceptibles e infectados, pero con 50 muertos más, es decir:  $P_f \approx (204.59, 0, 609.41)$

Bueno, vamos a intentar obtener esa formulita del tipo  $(\Delta S_f / \Delta S_0) = m(\nu, \beta)$

Primero declaramos la relación entre la pendiente y beta:

```
In [21]: N=400; gradsize=N
tiempo=np.linspace(0,100,100)
grad_S=range(N)
fig=plt.figure(figsize=(7,5))
import math
nu=0.4
grad_beta=np.linspace(0.0001,0.01,40)
pendientes_menlog=list()
for beta in grad_beta:
    array_Sf=list()
    array_SoN=list()
    for S in grad_S:
        P0=[S,N-S,0]
        Ps = scint.odeint(F, P0, tiempo)
        array_Sf.append(Ps[-1,0])
        array_SoN.append(S/N)
    m=(array_Sf[-1]-array_Sf[0])/(grad_S[-1]-grad_S[0])
    pendientes_menlog.append(-math.log(m))
relación=(pendientes_menlog[-1]-pendientes_menlog[-20])/(grad_beta[-1]-grad_beta[-20])
print('relación: ',relación)
print(r'$-\log(S_f/S_0)={}\cdot \beta$'.format(relación))
```

```
relación: 1006.1348864140069
$-log(S_f/S_0)=1006.1348864140069\cdot \beta$
<Figure size 700x500 with 0 Axes>
```

Está bien, ahora veamos la relación con nu:

```
In [ ]: import numpy as np
from scipy.optimize import curve_fit

# Datos observados:
x=list(grad_nu)
y=list(pendientes_m_nu)

# Definir la función logística
def logistic_function(x, L, k, x0):
    return L / (1 + np.exp(-k*(x-x0)))

# Realizar el ajuste de curva
params, _ = curve_fit(logistic_function, x, y)

# Obtener los parámetros estimados
L_estimado, k_estimado, x0_estimado = params

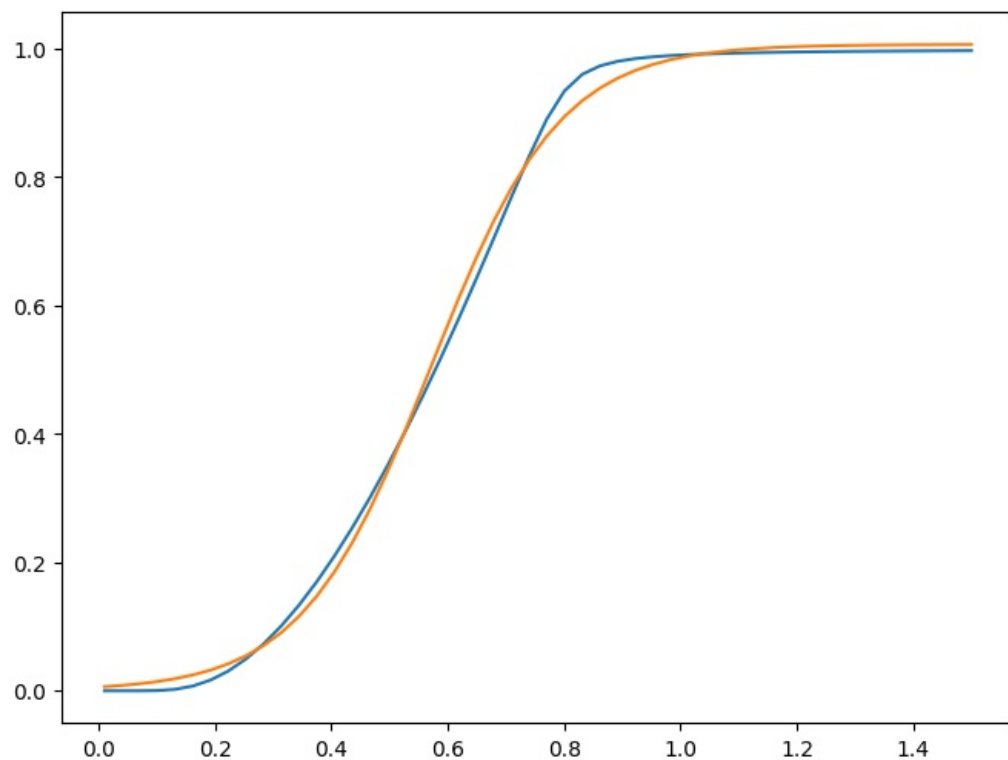
# Imprimir los parámetros estimados
print("Parámetro L estimado:", L_estimado)
print("Parámetro k estimado:", k_estimado)
print("Parámetro x0 estimado:", x0_estimado)
```

```
Parámetro L estimado: 1.0068349778913483
Parámetro k estimado: 9.014588213518607
Parámetro x0 estimado: 0.5704838325376685
```

Comparo la función real con la predicha:

```
In [ ]: fig=plt.figure(figsize=(8,6))
def logistica_predicha_nu(x):
    return L_estimado / (1 + np.exp(-k_estimado*(x-x0_estimado)))
plt.plot(grad_nu,pendientes_m_nu,label='curva real')
plt.plot(grad_nu,logistica_predicha_nu(grad_nu),label='curva predicha')
```

```
Out[ ]: [<matplotlib.lines.Line2D at 0x7f32c27f8ac0>]
```

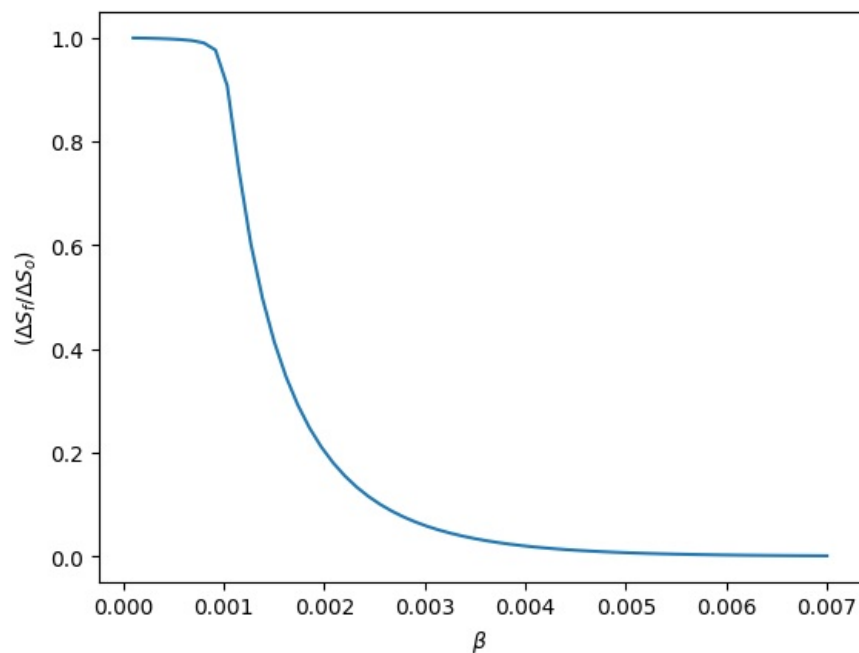


He cambiado de idea, voy a intentar trabajar con la sigmoidea entre  $(S_f/S_0)$  y  $\beta$ .

```
In [ ]: N=400; gradsize=N
tiempo=np.linspace(0,100,100)
grad_S=range(N)
nu=0.4
grad_beta=np.linspace(0.0001,0.007,60)
pendientes_m_beta=list()
for beta in grad_beta:
    array_Sf=list()
    array_SoN=list()
    for S in grad_S:
        P0=[S,N-S,0]
        Ps = scint.odeint(F, P0, tiempo)
        array_Sf.append(Ps[-1,0])
        array_SoN.append(S/N)
        m=(array_Sf[-1]-array_Sf[0])/(grad_S[-1]-grad_S[0])
        pendientes_m_beta.append(m)
plt.plot(grad_beta,pendientes_m_beta)
plt.xlabel(r'$\beta$')
plt.ylabel(r'$(\Delta S_f / \Delta S_o)$')
```

```
Out[ ]: Text(0, 0.5, '$(\Delta S_f / \Delta S_o)$')
```





estimo y comparo con la curva predicha...

```
In [ ]: # Datos observados:
x=list(grad_beta)
y=list(pendientes_m_beta)

# Realizar el ajuste de curva
params, _ = curve_fit(logistic_function, x, y)

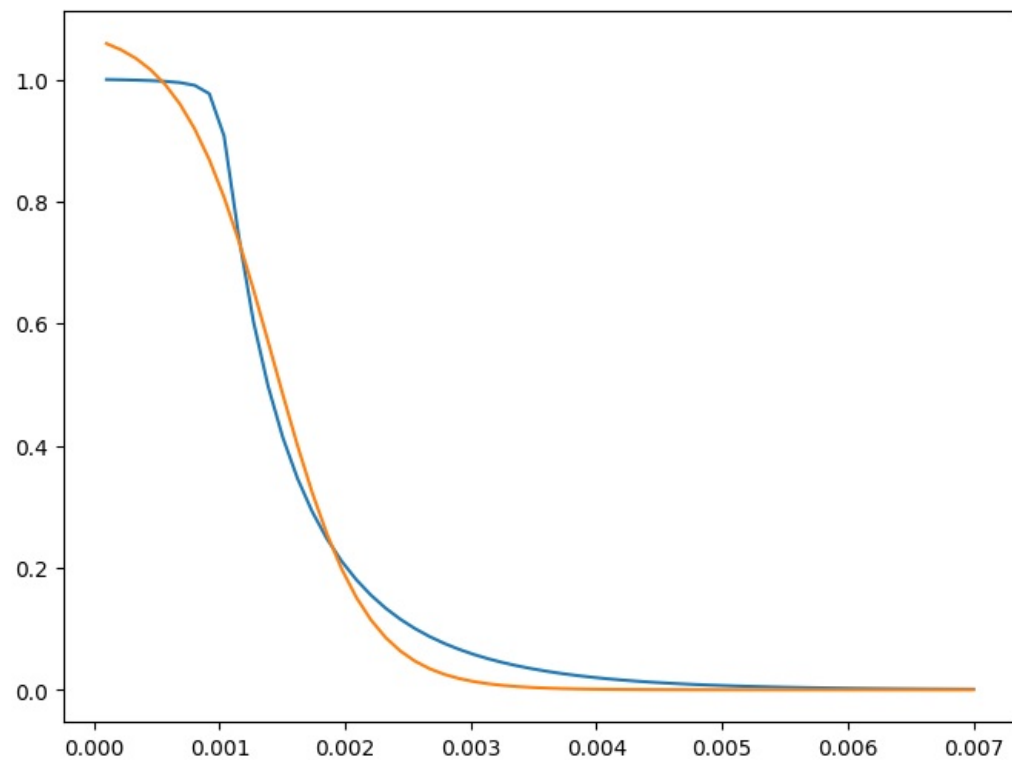
# Obtener los parámetros estimados
L_estimado, k_estimado, x0_estimado = params

# Imprimir los parámetros estimados
print("Parámetro L estimado:", L_estimado)
print("Parámetro k estimado:", k_estimado)
print("Parámetro x0 estimado:", x0_estimado)

#Veamos cómo se comparan
fig=plt.figure(figsize=(8,6))
def logistica_predicha_beta(x):
    return L_estimado / (1 + np.exp(-k_estimado*(x-x0_estimado)))
plt.plot(grad_beta,pendientes_m_beta,label='curva real')
plt.plot(grad_beta,logistica_predicha_beta(grad_beta),label='curva predicha')
```

```
Parámetro L estimado: 1.0870214171217352
Parámetro k estimado: -2734.9940198652253
Parámetro x0 estimado: 0.0014225957749670324
[<matplotlib.lines.Line2D at 0x7f32be044610>]
```

Out[ ]:



Es decir, obtengo una función que depende de  $\nu$  al fijar  $\beta$ ; y una función que depende de  $\beta$  al fijar  $\nu$ . Esto no me sirve, quiero una función que dependa de los dos parámetros sin importar su valor. El análisis se complica, nos estamos alejando del objetivo del trabajo... aquí concluye este cuaderno.

# MODELO SIRS

El modelo en un principio está regido por un sistema EDO de 3 ecuaciones tal que:

$$\frac{dS}{dt} = -\beta SI + \gamma R \quad \frac{dI}{dt} = \beta SI - \nu I \quad \frac{dR}{dt} = \nu I - \gamma R$$

Importamos los paquetes necesarios:

```
In [1]: import matplotlib.pyplot as plt
import scipy.integrate as scint
import sympy as sp
import numpy as np
sp.init_printing()
```

## Simulación numérica del modelo SIRS

Procedemos a una simulación en dos formatos: poblaciones vs. tiempo, y plano de fases susceptibles vs. infectados.

```
In [2]: beta=0.002; nu=0.4; gamma=0.02; N=750
def F(P, t):
    return [-beta*P[0]*P[1] + gamma*P[2], beta*P[0]*P[1]-nu*P[1], nu*P[1] - gamma*P[2]]
def F2(P, t):
    return [-beta*P[0]*P[1] + gamma*(N-P[1]-P[0]), beta*P[0]*P[1]-nu*P[1]]
tiempo = np.linspace(0, 70,100)
P0 = [730, 20, 0]
Ps = scint.odeint(F, P0, tiempo)

# fig=plt.figure(figsize=(12,6))
fig, axs = plt.subplots(1, 2, figsize=(11, 4), gridspec_kw={'width_ratios': [3, 2]})

# plt.subplot(121, title='Poblaciones vs. tiempo')
# plt.plot(tiempo, Ps[:,0], color="blue",label="Susceptibles")
# plt.plot(tiempo,Ps[:,1], color="red",label="Infectados")
# plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados")
# plt.legend()
# plt.xlabel('tiempo')
# plt.ylabel('hospedadores')

axs[0].scatter(0,P0[0],marker='o'); axs[0].scatter(0,P0[1],marker='o'); axs[0].scatter(0,P0[2],marker='o')
axs[0].plot(tiempo, Ps[:,0], color="blue",label="Susceptibles")
axs[0].plot(tiempo,Ps[:,1], color="red",label="Infectados")
axs[0].plot(tiempo,Ps[:,2],color="green",label="Recuperados")
axs[0].set_title('Poblaciones vs. tiempo')
axs[0].legend()
axs[0].set_xlabel('tiempo')
axs[0].set_ylabel('hospedadores')

# plt.subplot(122, title='plano de fases')
tiempo2 = np.linspace(0, 100,100)
# for j in range (1,4):
#     P02 = [100, j*10]
#     Ps2 = scint.odeint(F2, P02, tiempo2)
#     plt.plot(Ps2[:,0],Ps2[:,1])
#     plt.xlabel("Susceptibles")
#     plt.ylabel("Infectados")
# for j in range (0,3):
#     P02 = [100+20*j, 60]
#     Ps2 = scint.odeint(F2, P02, tiempo2)
#     plt.plot(Ps2[:,0],Ps2[:,1])
#     plt.xlabel("Susceptibles")
#     plt.ylabel("Infectados")
# plt.legend()
# plt.xlabel('susceptibles')
# plt.ylabel('infectados')

for j in range (1,5):
    P02 = [100, j*10]
    Ps2 = scint.odeint(F2, P02, tiempo2)
    axs[1].scatter(P02[0], P02[1], marker='o')
    axs[1].plot(Ps2[:,0],Ps2[:,1])
```

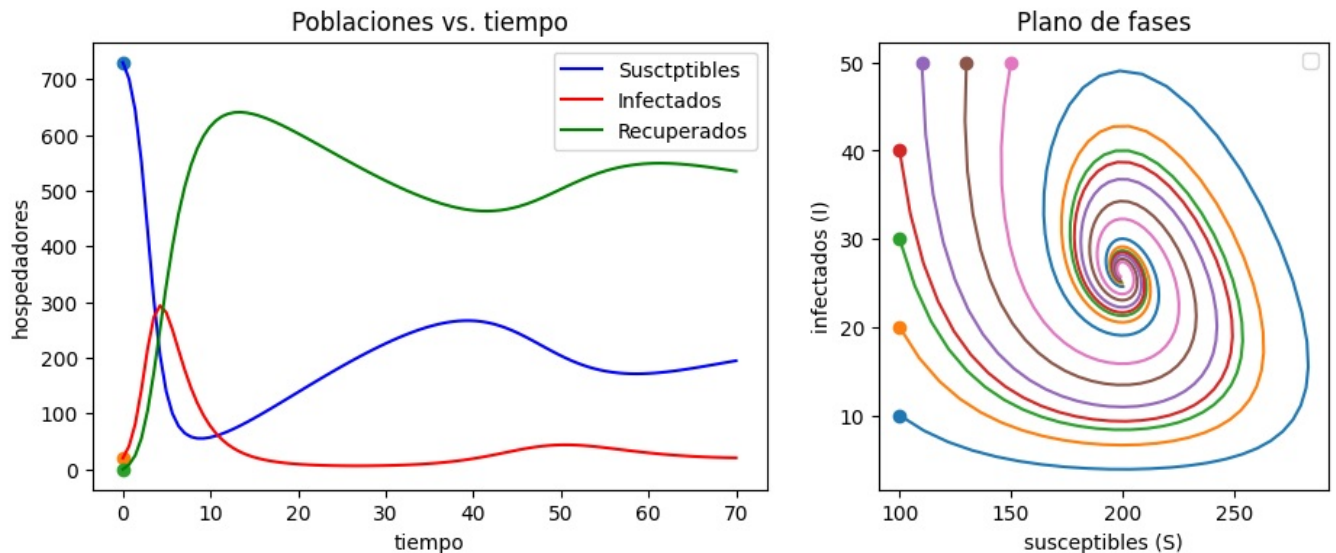
```

for j in range(0,3):
    P02 = [110+20*j, 50]
    Ps2 = scint.odeint(F2, P02, tiempo2)
    axs[1].scatter(P02[0], P02[1], marker='o')
    axs[1].plot(Ps2[:,0],Ps2[:,1])
axs[1].set_title('Plano de fases')
axs[1].set_xlabel("susceptibles (S)")
axs[1].set_ylabel("infectados (I)")
axs[1].legend()

# fig.suptitle(r'Modelo SIRS: $\beta=0.002$ $\nu=0.4$ $\gamma=0.02$ $N=1000$', fontsize=16)
# plt.subplots_adjust(top=0.8)
plt.savefig('First_view_SIRS.png')
plt.show()

```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



## Estudio cualitativo del modelo SIRS

Para realizar el estudio cualitativo del sistema por los métodos vistos en clase, necesitamos reducir el sistema de 3 a 2 ecuaciones diferenciales. Al ser un sistema cerrado, se cumple que  $N = S + I + R$ , de modo que podemos quitarnos la variable  $R$  sustituyéndola por  $N - S - I$ . Con esto, nos queda el sistema:

$$\begin{cases} S'(t) = -\beta SI + \gamma(N - S - I) \\ I'(t) = \beta SI - \nu I \end{cases}$$

Para comprobar que esto no afecta al comportamiento del sistema, hacemos una breve representación gráfica:

```

In [3]: beta=0.002; nu=0.4; gamma=0.02; N=750
tiempo = np.linspace(0, 100, 100)
P0 = [730, 20, 0]
Ps = scint.odeint(F, P0, tiempo)
P02 = [730, 20]
Ps2 = scint.odeint(F2, P02, tiempo)

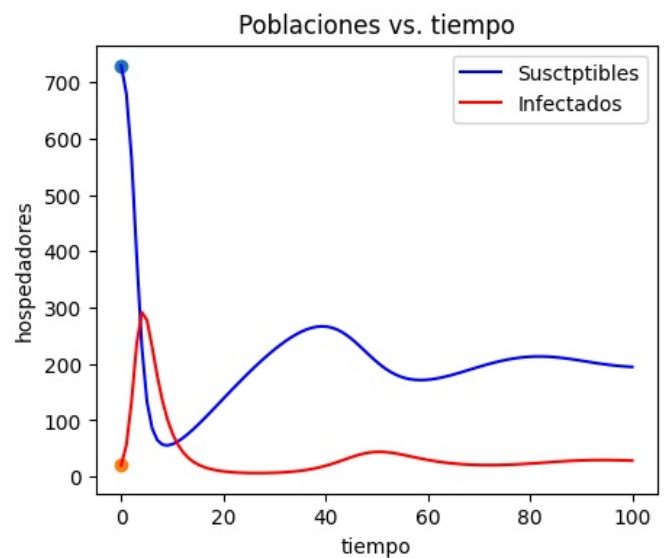
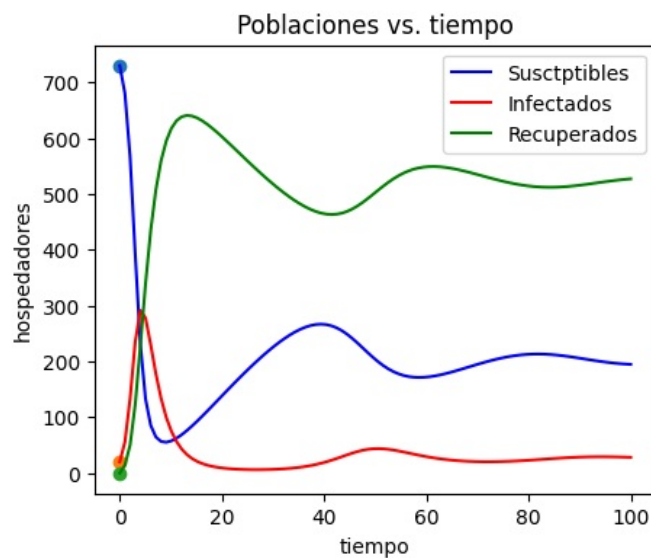
fig, axs = plt.subplots(1, 2, figsize=(11, 4))

axs[0].scatter(0,P0[0],marker='o'); axs[0].scatter(0,P0[1],marker='o'); axs[0].scatter(0,P0[2],marker='o')
axs[0].plot(tiempo, Ps[:,0], color="blue",label="Susceptibles")
axs[0].plot(tiempo,Ps[:,1], color="red",label="Infectados")
axs[0].plot(tiempo,Ps[:,2],color="green",label="Recuperados")
axs[0].set_title('Poblaciones vs. tiempo')
axs[0].legend()
axs[0].set_xlabel('tiempo')
axs[0].set_ylabel('hospedadores')

axs[1].scatter(0,P02[0],marker='o'); axs[1].scatter(0,P02[1],marker='o')
axs[1].plot(tiempo2, Ps2[:,0], color="blue",label="Susceptibles")
axs[1].plot(tiempo2,Ps2[:,1], color="red",label="Infectados")
axs[1].set_title('Poblaciones vs. tiempo')
axs[1].legend()
axs[1].set_xlabel('tiempo')
axs[1].set_ylabel('hospedadores')

plt.subplots_adjust()
plt.show()

```



Como se observa, la eliminación de la variable "R" no altera el comportamiento de S y de I

Reanudamos pues el estudio cualitativo usando el módulo sympy:

```
In [4]: s, i, r = sp.symbols("S I R")
        beta, gamma, n, v = sp.symbols("beta gamma N nu", positive=True)
        fsi = -beta*s*i + gamma*(n-s-i)
        gsi = beta*s*i - v*i
        r = n-s-i
        fsi, gsi
```

```
Out[4]: (- I S β + γ(- I + N - S), I S β - I v)
```

```
In [5]: p_cr = sp.solve([fsi, gsi], [s, i], simplify=True)
        p_cr
```

```
Out[5]: [(N, 0), (v β, γ(N β - v) β (γ + v))]
```

```
In [6]: par_matrix = sp.Matrix([[fsi.diff(s), fsi.diff(i)], [gsi.diff(s), gsi.diff(i)]])
        par_matrix
```

```
Out[6]: [- I β - γ  - S β - γ I β  S β - v]
```

Para el primer punto de equilibrio

```
In [7]: C1 = par_matrix.subs({s: p_cr[0][0], i: p_cr[0][1]})
        C1
```

```
Out[7]: [- γ  - N β - γ 0  N β - v]
```

```
In [8]: Clauto = C1.eigenvals()
        Clautovect = C1.eigenvects()
        Clauto, Clautovect
```

```
Out[8]: ({- γ: 1, N β - v: 1}) [(- γ, 1), (N β - v, 1)]
```

```
In [9]: trazaC1 = sp.simplify(sp.trace(C1))
        detC1 = sp.simplify(sp.det(C1))
        trazaC1, detC1
```

```
Out[9]: (N β - γ - v, γ(- N β + v))
```

Primero punto crítico

Condición de estabilidad asintótica del sistema lineal:  $\text{tr}(A) < 0$  ;  $\det(A) > 0$

Esto es:

$$\begin{cases} N\beta - \gamma - \nu < 0 \\ \gamma(-N\beta + \nu) > 0 \end{cases}$$

Sabiendo que los parámetros del modelo ( $\gamma, \nu, \beta$  y  $N$ ) son positivos, vemos que la estabilidad de este punto crítico depende de si  $(N\beta - \nu < 0)$ , es decir, de si  $R_0 < 1$ , donde  $R_0 = (N\beta)/\nu$ . Si  $R_0 < 1$ , entonces  $\text{tr}(A) < 0$  y  $\det(A) > 0$ , por lo que el punto sería asintóticamente estable. En cambio, si  $R_0 > 1$ , entonces  $\det(A) < 0$  y, por tanto, el punto se comporta como inestable. Cuando  $R_0 = 1$ , tenemos que  $\text{tr}(A) < 0$  y  $\det(A) = 0$ , no podemos conocer a priori la estabilidad. Para comprobarlo, podemos graficar los tres casos para varios PVI.

```
In [10]: fig=plt.figure("figuraprueba",figsize=(16,5))
fig.suptitle("funciones")
tiempo2 = np.linspace(0, 500,500)

# La primera gráfica de dos en una fila con dos columnas
plt.subplot(131, title = '$R_{0}>1$')
beta=0.0022;nu=0.447; gamma=0.02; N=1000
for j in range (1,7):
    P02 = [10, j*10]
    Ps2 = scint.odeint(F2, P02, tiempo2)
    plt.plot(Ps2[:,0],Ps2[:,1])
    plt.xlabel("Susceptibles")
    plt.ylabel("Infectados")

plt.xlabel('S')
plt.ylabel('I')

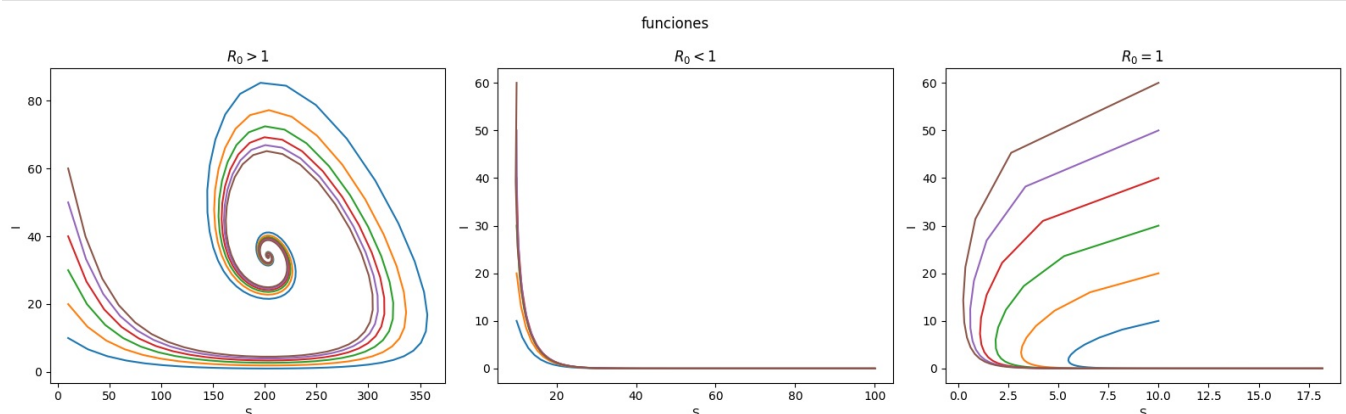
# La segunda gráfica
plt.subplot(132, title = '$R_{0}<1$')
beta=0.0022;nu=0.447; gamma=0.02; N=100
for j in range (1,7):
    P02 = [10, j*10]
    Ps2 = scint.odeint(F2, P02, tiempo2)
    plt.plot(Ps2[:,0],Ps2[:,1])
    plt.xlabel("Susceptibles")
    plt.ylabel("Infectados")

plt.xlabel('S')
plt.ylabel('I')
# No ponemos la 'y' para que no se solape

# tercera gráfica??
plt.subplot(133, title = '$R_{0}=1$')
beta=0.022;nu=0.4; gamma=0.02; N=nu/beta
for j in range (1,7):
    P02 = [10, j*10]
    Ps2 = scint.odeint(F2, P02, tiempo2)
    plt.plot(Ps2[:,0],Ps2[:,1])
    plt.xlabel("Susceptibles")
    plt.ylabel("Infectados")

plt.xlabel('S')
plt.ylabel('I')

fig.tight_layout()
plt.show()
```



Vemos que el comportamiento de estabilidad asintótica es el mismo tanto para  $R_0 < 1$  como para  $R_0 = 1$ . Por lo tanto, concluimos que el punto  $(N,0,0)$  sólo es inestable cuando  $R_0 > 1$ .

Nótese que, si partimos de  $S=N-R$ ,  $I=0$  y  $R=N-S$ , da igual qué valores tomen S y R que al final todo R irá hacia S y acabaremos en

(N,0,0), independientemente de los valores de  $\gamma$ , N y  $\nu$ , es decir, independientemente de  $R_0$ . Veamos una prueba:

```
In [11]: # Modulos con los que trabajaremos
from mpl_toolkits.mplot3d import axes3d

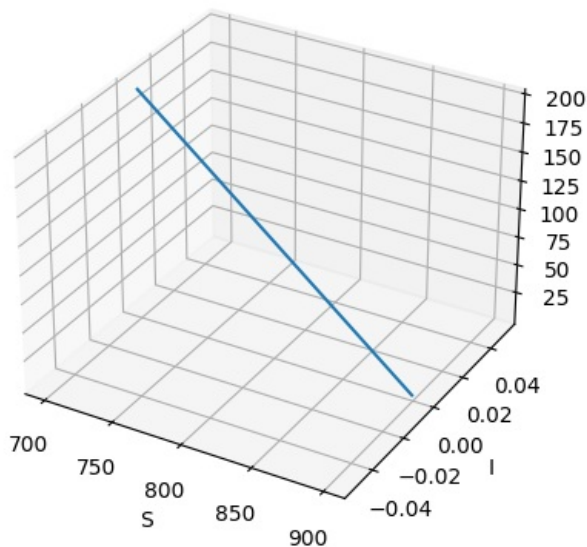
# Creamos la figura
fig = plt.figure()

# Agrrgamos un plano 3D
ax1 = fig.add_subplot(111,projection='3d')

# Datos en array bi-dimensional
beta=0.0022;nu=0.447; gamma=0.1
def F(P, t):
    return [-beta*P[0]*P[1] + gamma*P[2], beta*P[0]*P[1]-nu*P[1], nu*P[1] - gamma*P[2]]
tiempo3 = np.linspace(0, 100,100)
P03 = [700, 0, 200]
Ps3 = scint.odeint(F, P03, tiempo3)
# plt.plot(tiempo, Ps[:,0], color="blue",label="Susctptibles")
# plt.plot(tiempo,Ps[:,1], color="red",label="Infectados")
# plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados")
# plt.legend()
x=np.array([list(Ps3[:,0])])
y=np.array([list(Ps3[:,1])])
z=np.array([list(Ps3[:,2])])

# plot_wireframe nos permite agregar los datos x, y, z. Por ello 3D
# Es necesario que los datos esten contenidos en un array bi-dimensional
ax1.plot_wireframe(x,y,z)
#Ponemos nombres a los ejes:
ax1.set_xlabel('S')
ax1.set_ylabel('I')
ax1.set_zlabel('R')

# Mostramos el gráfico
plt.show()
```



Lo que observamos es que partimos de [700,0,200] y tendemos hacia [900,0,0] (no llegamos nunca a él porque el decaimiento de los recuperados es exponencial...). Entonces, en este caso, los parámetros indican que  $R_0 > 1$ , luego el punto debería ser inestable. Sin embargo, vemos que se comporta como asintóticamente estable. Nuestra hipótesis es que esto no es relevante. La estabilidad del punto (N,0,0) depende de  $R_0$ . En los casos en los que no tenga sentido hablar de  $R_0$  (no hay infectados, no hay infección), no tiene sentido analizar la estabilidad del punto de la misma forma. Los PVI correspondientes a (S,0,N-S) para todo S, tal que  $0 \leq S \leq N$ , representan un conjunto concreto de casos en los que parte de la población se encuentra inmunizada para una enfermedad, pero la enfermedad no está presente, por lo que toda la subpoblación R pierde la inmunidad con el tiempo y tendemos hacia el punto (N,0,0).

## Para el segundo punto critico

```
In [12]: C2=par_matrix.subs({s:p_cr[1][0],i:p_cr[1][1]})
C2=sp.simplify(C2)
C2
```

Out[12]:

$$\begin{bmatrix} \gamma(-N\beta - \gamma)\gamma + v & -\gamma - v\gamma(N\beta - v)\gamma + v \\ 0 & 0 \end{bmatrix}$$

```
In [13]: C2auto=sp.simplify(C2.eigenvals())
sp.simplify(C2auto)
```

Out[13]:

$$\left[ -\sqrt{\gamma}\sqrt{N^2\beta^2\gamma - 2N\beta\gamma^2 - 8N\beta\gamma v - 4N\beta v^2 + \gamma^3 + 4\gamma^2v + 8\gamma v^2 + 4v^3}2(\gamma + v) - \gamma(N\beta + \gamma)2(\gamma + v); 1, \sqrt{\gamma}\sqrt{N^2\beta^2\gamma - 2N\beta\gamma^2 - 8N\beta\gamma v - 4N\beta v^2 + \gamma^3 + 4\gamma^2v + 8\gamma v^2 + 4v^3}2(\gamma + v) - \gamma(N\beta + \gamma)2(\gamma + v) \right]$$

```
In [14]: trazaC2=sp.simplify(sp.trace(C2))
detC2=sp.simplify(sp.det(C2))
trazaC2, detC2
```

Out[14]:

$$\left( -\gamma(N\beta + \gamma)\gamma + v, \gamma(N\beta - v) \right)$$

Condición de estabilidad asintótica del sistema lineal:  $\text{tr}(A) < 0$  ;  $\det(A) > 0$

Esto es:

$$\begin{bmatrix} \gamma(-N\beta - \gamma)\gamma + v < 0 \\ \gamma(N\beta - v) > 0 \end{bmatrix}$$

Los parámetros del modelo ( $\gamma$ ,  $v$ ,  $\beta$ ) son positivos, en este caso siempre  $\text{tr}(A) < 0$ . En cuanto al determinante, tres casos:  $R_0 > 1 \rightarrow$

$\det(A) > 0 \rightarrow$  Estabilidad asintótica

$R_0 < 1 \rightarrow \det(A) < 0 \rightarrow$  Inestabilidad

$R_0 = 1 \rightarrow \det(A) = 0 \rightarrow$  Mismo caso de antes. Inestabilidad.

Es decir, este punto será asintóticamente estable siempre que  $N\beta - v > 0$ , es decir,  $R_0 > 1$ .

## Dinámica de los puntos críticos

Veamos cómo es el sistema para los tres casos de  $R_0$  y para varias condiciones iniciales:

```
In [15]: import matplotlib.pyplot as plt
fig=plt.figure(figsize=(13,4.5))
tiempo2 = np.linspace(0, 200,2000)
pm=20
X, Y = np.meshgrid(np.linspace(0, 750, pm), np.linspace(0, 60, pm))

beta=0.002; N=750; nu=0.6; gamma=0.02; R0=(N*beta)/nu
plt.subplot(131)
U = -beta*X*Y + gamma*(N-X-Y)
V = beta*X*Y - nu*Y
modulo2 = U**2 + V**2
D1 = np.ones([pm, pm])
D2 = np.ones([pm, pm])
for i in range(pm):
    for j in range(pm):
        if modulo2[i, j] == 0:
            D1[i, j] = 0
            D2[i, j] = 0
        else:
            D1[i, j] = U[i, j]/np.sqrt(modulo2[i, j])
            D2[i, j] = V[i, j]/np.sqrt(modulo2[i, j])
plt.quiver(X, Y, D1, D2, scale=35)
for j in [[10,20],[10,60],[60,60],[120,60]]:
    P02 = j
    Ps2 = scint.odeint(F2, P02, tiempo2)
    plt.plot(Ps2[:,0],Ps2[:,1])
    plt.xlabel("Susceptibles")
    plt.ylabel("Infectados")
PE1_1=[N,0]
PE2_1=[round((nu/beta),2),round((gamma*(N*beta-nu))/(beta*(gamma+nu)),2)]
```



```

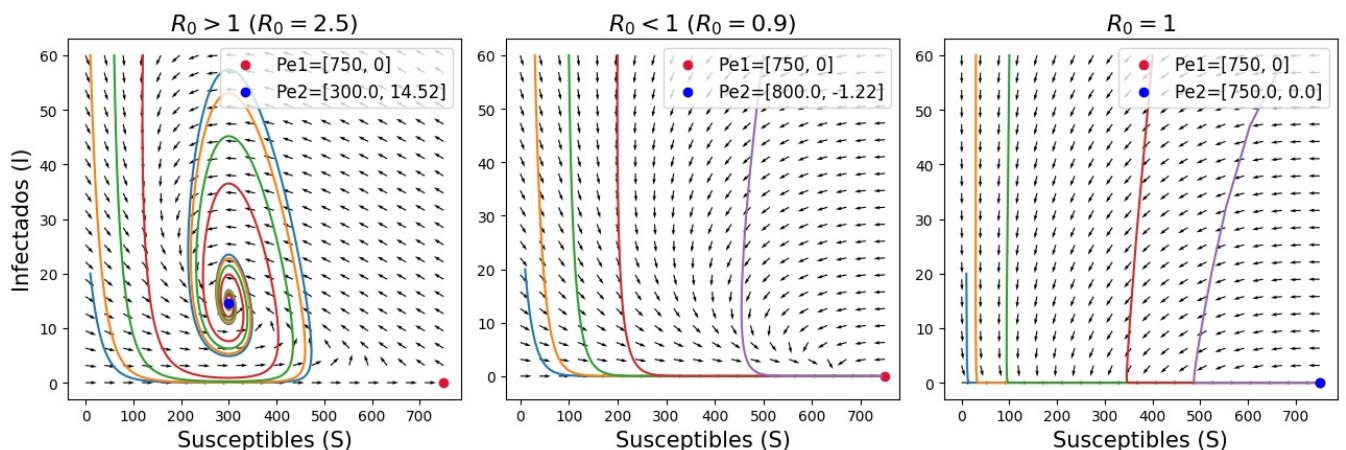
plt.scatter(PE1_1[0],PE1_1[1],color='crimson',label='Pe1={}'.format(PE1_1))
plt.scatter(PE2_1[0],PE2_1[1],color='blue',label='Pe2={}'.format(PE2_1),zorder=2)
plt.xlabel("Susceptibles (S)",fontsize=15)
plt.ylabel("Infectados (I)",fontsize=15)
plt.title(r'$R_0>1$ ($R_0=${})'.format(round(R0,1)),fontsize=16)
plt.legend(fontsize='large')

beta=0.001; N=750; nu=0.8; gamma=0.02; R0=(N*beta)/nu
plt.subplot(132)
U = -beta*X*Y + gamma*(N-X-Y)
V = beta*X*Y - nu*Y
modulo2 = U**2 + V**2
D1 = np.ones([pm, pm])
D2 = np.ones([pm, pm])
for i in range(pm):
    for j in range(pm):
        if modulo2[i, j] == 0:
            D1[i, j] = 0
            D2[i, j] = 0
        else:
            D1[i, j] = U[i, j]/np.sqrt(modulo2[i, j])
            D2[i, j] = V[i, j]/np.sqrt(modulo2[i, j])
plt.quiver(X, Y, D1, D2, scale=35)
for j in [[10,20],[30,60],[100,60],[200,60],[500,60]]:
    P02 = j
    Ps2 = scint.odeint(F2, P02, tiempo2)
    plt.plot(Ps2[:,0],Ps2[:,1])
PE1_2=[N,0]
PE2_2=[round((nu/beta),2),round((gamma*(N*beta-nu))/(beta*(gamma+nu)),2)]
plt.scatter(PE1_2[0],PE1_2[1],color='crimson',label='Pe1={}'.format(PE1_2))
plt.scatter(PE2_2[0],PE2_2[1],color='blue',label='Pe2={}'.format(PE2_2),zorder=2)
plt.xlim(-30,780)
plt.xlabel("Susceptibles (S)",fontsize=15)
plt.title(r'$R_0<1$ ($R_0=${})'.format(round(R0,1)),fontsize=16)
plt.legend(loc='upper right',fontsize='large')

tiempo2 = np.linspace(0, 1000,10000)
plt.subplot(133)
beta=0.022; N=750 ;nu=N*beta; gamma=0.02
U = -beta*X*Y + gamma*(N-X-Y)
V = beta*X*Y - nu*Y
modulo2 = U**2 + V**2
D1 = np.ones([pm, pm])
D2 = np.ones([pm, pm])
for i in range(pm):
    for j in range(pm):
        if modulo2[i, j] == 0:
            D1[i, j] = 0
            D2[i, j] = 0
        else:
            D1[i, j] = U[i, j]/np.sqrt(modulo2[i, j])
            D2[i, j] = V[i, j]/np.sqrt(modulo2[i, j])
plt.quiver(X, Y, D1, D2, scale=35)
for j in [[10,20],[30,60],[100,60],[400,60],[680,60]]:
    P02 = j
    Ps2 = scint.odeint(F2, P02, tiempo2)
    plt.plot(Ps2[:,0],Ps2[:,1])
PE1_3=[N,0]
PE2_3=[round((nu/beta),2),round((gamma*(N*beta-nu))/(beta*(gamma+nu)),2)]
plt.scatter(PE1_3[0],PE1_3[1],color='crimson',label='Pe1={}'.format(PE1_3))
plt.scatter(PE2_3[0],PE2_3[1],color='blue',label='Pe2={}'.format(PE2_3),zorder=2)
plt.legend(loc='upper right',fontsize='large')
plt.xlabel("Susceptibles (S)",fontsize=15)
plt.title(r'$R_0=1$',fontsize=16)

fig.tight_layout()
plt.show()

```



La pregunta es ¿a qué valores tienden las poblaciones cuando  $R_0 = 1$ ?

La pregunta es ¿a qué valores pertenecen las poblaciones cuando  $R_0 > 1$ ?

```
In [16]: print (Ps2[-1,:])

[ 7.50000003e+02 -4.52343355e-09]
```

Campo de direcciones entorno al punto (N,0,0) para distintos  $R_0$

```
In [36]: import matplotlib.pyplot as plt
fig=plt.figure(figsize=(13,4.5))
tiempo2 = np.linspace(0, 200,2000)
pm=25
X, Y = np.meshgrid(np.linspace(450, 1050, pm), np.linspace(-60, 60, pm))

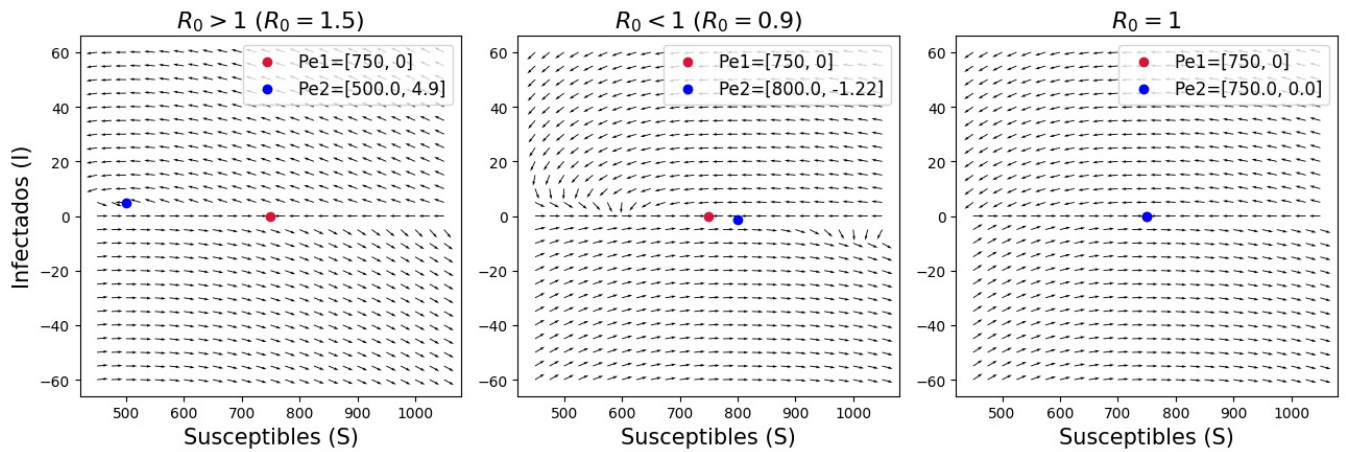
beta=0.002; N=750; nu=1; gamma=0.02; R0=(N*beta)/nu
plt.subplot(131)
U = -beta*X*Y + gamma*(N-X-Y)
V = beta*X*Y - nu*Y
modulo2 = U**2 + V**2
D1 = np.ones([pm, pm])
D2 = np.ones([pm, pm])
for i in range(pm):
    for j in range(pm):
        if modulo2[i, j] == 0:
            D1[i, j] = 0
            D2[i, j] = 0
        else:
            D1[i, j] = U[i, j]/np.sqrt(modulo2[i, j])
            D2[i, j] = V[i, j]/np.sqrt(modulo2[i, j])
plt.quiver(X, Y, D1, D2, scale=35)
PE1_1=[N,0]
PE2_1=[round((nu/beta),2),round((gamma*(N*beta-nu))/(beta*(gamma+nu)),2)]
plt.scatter(PE1_1[0],PE1_1[1],color='crimson',label='Pe1={}'.format(PE1_1))
plt.scatter(PE2_1[0],PE2_1[1],color='blue',label='Pe2={}'.format(PE2_1),zorder=2)
plt.xlabel("Susceptibles (S)",fontsize=15)
plt.ylabel("Infectados (I)",fontsize=15)
plt.title(r'$R_0>1$ ($R_0=${})'.format(round(R0,1)),fontsize=16)
plt.legend(fontsize='large', loc='upper right')

beta=0.001; N=750; nu=0.8; gamma=0.02; R0=(N*beta)/nu
plt.subplot(132)
U = -beta*X*Y + gamma*(N-X-Y)
V = beta*X*Y - nu*Y
modulo2 = U**2 + V**2
D1 = np.ones([pm, pm])
D2 = np.ones([pm, pm])
for i in range(pm):
    for j in range(pm):
        if modulo2[i, j] == 0:
            D1[i, j] = 0
            D2[i, j] = 0
        else:
            D1[i, j] = U[i, j]/np.sqrt(modulo2[i, j])
            D2[i, j] = V[i, j]/np.sqrt(modulo2[i, j])
plt.quiver(X, Y, D1, D2, scale=35)
PE1_2=[N,0]
PE2_2=[round((nu/beta),2),round((gamma*(N*beta-nu))/(beta*(gamma+nu)),2)]
plt.scatter(PE1_2[0],PE1_2[1],color='crimson',label='Pe1={}'.format(PE1_2))
plt.scatter(PE2_2[0],PE2_2[1],color='blue',label='Pe2={}'.format(PE2_2),zorder=2)
# plt.xlim(-30,780)
plt.xlabel("Susceptibles (S)",fontsize=15)
plt.title(r'$R_0<1$ ($R_0=${})'.format(round(R0,1)),fontsize=16)
plt.legend(loc='upper right',fontsize='large')

tiempo2 = np.linspace(0, 1000,10000)
plt.subplot(133)
beta=0.022; N=750 ;nu=N*beta; gamma=0.02
U = -beta*X*Y + gamma*(N-X-Y)
V = beta*X*Y - nu*Y
modulo2 = U**2 + V**2
D1 = np.ones([pm, pm])
D2 = np.ones([pm, pm])
for i in range(pm):
    for j in range(pm):
        if modulo2[i, j] == 0:
            D1[i, j] = 0
            D2[i, j] = 0
        else:
            D1[i, j] = U[i, j]/np.sqrt(modulo2[i, j])
            D2[i, j] = V[i, j]/np.sqrt(modulo2[i, j])
plt.quiver(X, Y, D1, D2, scale=35)
PE1_3=[N,0]
PE2_3=[round((nu/beta),2),round((gamma*(N*beta-nu))/(beta*(gamma+nu)),2)]
plt.scatter(PE1_3[0],PE1_3[1],color='crimson',label='Pe1={}'.format(PE1_3))
plt.scatter(PE2_3[0],PE2_3[1],color='blue',label='Pe2={}'.format(PE2_3),zorder=2)
plt.legend(loc='upper right',fontsize='large')
plt.xlabel("Susceptibles (S)",fontsize=15)
```

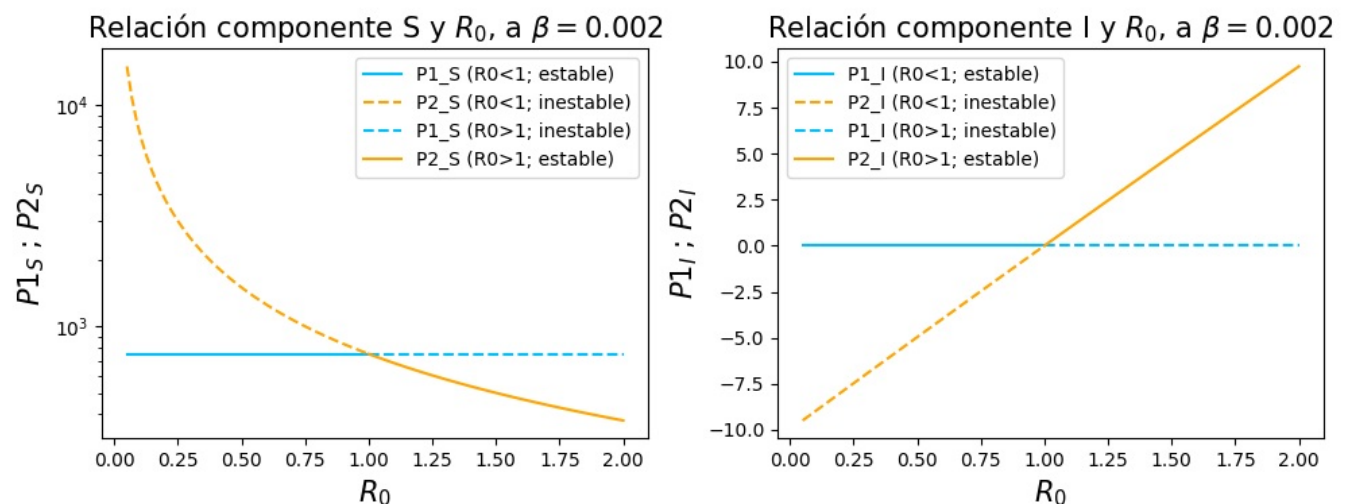
```
plt.title(r'$R_0=1$', fontsize=16)
```

```
fig.tight_layout()
plt.show()
```



## Bifurcación transcítica del modelo SIRS

```
In [17]: fig=plt.figure(figsize=(10,4))
N=750;beta=0.002;gamma=0.02
grad_R0_1=np.linspace(0.05,1,50)
grad_R0_2=np.linspace(1,2,50)
grad_nu_1=(N*beta)/grad_R0_1
grad_nu_2=(N*beta)/grad_R0_2
plt.subplot(121)
plt.plot(grad_R0_1,(N+0*grad_R0_1),linestyle='--',color='deepskyblue',label=r'P1_S (R0<1; estable)')
plt.plot(grad_R0_1,(grad_nu_1/beta),linestyle='--',color='orange',label=r'P2_S (R0<1; inestable)')
plt.plot(grad_R0_2,(N+0*grad_R0_1),linestyle='--',color='deepskyblue',label=r'P1_S (R0>1; inestable)')
plt.plot(grad_R0_2,(grad_nu_2/beta),linestyle='--',color='orange',label=r'P2_S (R0>1; estable)')
plt.ylabel(r'$P1_S$ ; $P2_S$', fontsize=15)
plt.xlabel(r'$R_0$', fontsize=15)
plt.legend(fontsize='medium')
plt.title(r'Relación componente S y $R_0$, a $\beta=0.002$', fontsize=15)
plt.yscale('log')
plt.subplot(122)
plt.plot(grad_R0_1,(0*grad_R0_1),linestyle='--',color='deepskyblue',label=r'P1_I (R0<1; estable)')
plt.plot(grad_R0_1,(gamma*(N*beta-grad_nu_1))/(beta*(gamma+grad_nu_1)),linestyle='--',color='orange',label=r'P2_I (R0<1; inestable)')
plt.plot(grad_R0_2,(0*grad_R0_1),linestyle='--',color='deepskyblue',label=r'P1_I (R0>1; inestable)')
plt.plot(grad_R0_2,(gamma*(N*beta-grad_nu_2))/(beta*(gamma+grad_nu_2)),linestyle='--',color='orange',label=r'P2_I (R0>1; estable)')
plt.xlabel(r'$R_0$', fontsize=15)
plt.ylabel(r'$P1_I$ ; $P2_I$', fontsize=15)
plt.legend(fontsize='medium')
plt.title(r'Relación componente I y $R_0$, a $\beta=0.002$', fontsize=15)
fig.tight_layout()
plt.show()
```



\*Ejercicio 25 del libro: SIRS con dinámicas vitales

$$\frac{dS}{dt} = -\beta SI + \gamma R + \delta N - \delta S \quad \frac{dI}{dt} = \beta SI - \nu I - \delta I \quad \frac{dR}{dt} = \nu I - \gamma R - \delta R$$

```
In [18]: s, i, r = sp.symbols("S I R")
b, g, d, n, N = sp.symbols("beta gamma delta nu N", positive=True)
fsi = -b*s*i + g*(N-s-i) + d*N-d*s
gsi = b*s*i - i*n - i*d
hsi = n*i - r*(g+d)
r = N-s-i
fsi, gsi, hsi
```

```
Out[18]: (- ISβ + Nδ - Sδ + γ(- I + N - S), ISβ - Iδ - Iv, Iv - R(δ + γ))
```

```
In [19]: p_cr_2 = sp.solve((fsi, gsi), (s, i))
p_cr_2
```

```
Out[19]:
```

$$\begin{bmatrix} -N\delta - N\gamma\delta + \gamma, 0 \\ -N\delta - N\gamma + (\delta + \gamma)(N\beta - \delta - \nu)\beta\delta + \gamma, (\delta + \gamma)(N\beta - \delta - \nu)\beta(\delta + \gamma + \nu) \end{bmatrix}$$

```
In [20]: par_matrix = sp.Matrix([[fsi.diff(s), fsi.diff(i)], [gsi.diff(s), gsi.diff(i)]])
par_matrix
```

```
Out[20]:
```

$$\begin{bmatrix} -I\beta - \delta - \gamma & -S\beta - \gamma I\beta & S\beta - \delta - \nu \end{bmatrix}$$

Primer punto de equilibrio

```
In [21]: C1 = par_matrix.subs({s: p_cr_2[0][0], i: p_cr_2[0][1]})
C1 = sp.simplify(C1)
C1
```

```
Out[21]:
```

$$\begin{bmatrix} -\delta - \gamma & -N\beta - \gamma\delta & N\beta - \delta - \nu \end{bmatrix}$$

```
In [22]: Clauto = C1.eigenvals()
Clautovect = C1.eigenvects()
Clauto, Clautovect
```

```
Out[22]:
```

$$\left( \{-\delta - \gamma: 1, N\beta - \delta - \nu: 1\} \left( -\delta - \gamma, 1, 10 \right) \left( N\beta - \delta - \nu, 1, -N\beta - \gamma N\beta + \gamma - \nu \right) \right)$$

```
In [23]: trazaC1 = sp.simplify(sp.trace(C1))
detC1 = sp.simplify(sp.det(C1))
trazaC1, detC1
```

```
Out[23]:
```

$$\left( N\beta - 2\delta - \gamma - \nu, -N\beta\delta - N\beta\gamma + \delta^2 + \delta\gamma + \delta\nu + \gamma\nu \right)$$

Segundo punto crítico

```
In [24]: C2 = par_matrix.subs({s: p_cr_2[1][0], i: p_cr_2[1][1]})
C2 = sp.simplify(C2)
C2
```

```
Out[24]:
```

$$\begin{bmatrix} -N\beta\delta - N\beta\gamma - \delta\gamma - \gamma^2\delta + \gamma\nu & -\delta - \gamma - \nu - (\delta + \gamma)(-N\beta + \delta + \nu)\delta + \gamma\nu \\ 0 & 0 \end{bmatrix}$$

```
In [25]: C2auto = C2.eigenvals()
```

```
C2auto=C2.eigenvecs(),
C2autovect=C2.eigenvecs()
C2auto, C2autovect
```

Out[25]:

$$\left\{ -\sqrt{\delta} + \gamma \sqrt{N^2 \beta^2 \delta + N^2 \beta^2 \gamma - 4N\beta\delta^2 - 6N\beta\delta\gamma - 8N\beta\delta v - 2N\beta\gamma^2 - 8N\beta\gamma v - 4N\beta v^2 + 4\delta^3 + 8\delta^2\gamma + 12\delta^2 v + 5\delta\gamma^2 + 16\delta\gamma v + 12\delta v^2 + \gamma^3 + 4} \right.$$

```
In [26]: trazaC2=sp.simplify(sp.trace(C2))
detC2=sp.simplify(sp.det(C2))
trazaC2, detC2
```

Out[26]:

$$\left( -N\beta\delta - N\beta\gamma - \delta\gamma - \gamma^2\delta + \gamma + v, N\beta\delta + N\beta\gamma - \delta^2 - \delta\gamma - \delta v - \gamma v \right)$$

Processing math: 100%

# MODELO SIRV

El modelo en un principio está regido por un sistema EDO de 3 ecuaciones tal que:

$$\begin{cases} \frac{dS}{dt} = -\beta SI - V(t)S \\ \frac{dI}{dt} = \beta SI - \nu I \\ \frac{dR}{dt} = \nu I + V(t)S \end{cases}$$

En este cuaderno consideramos la hipótesis:  $V(t) = \zeta$

El hecho de establecer la tasa de vacunación  $V(t)$  en función de la población de susceptibles  $S$  no tiene tanto sentido lógico, puesto que en la vida real no se vacuna menos gente conforme hay menos gente susceptible. En cambio, esta aproximación proporciona un mayor sentido biológico.

## Simulación numérica

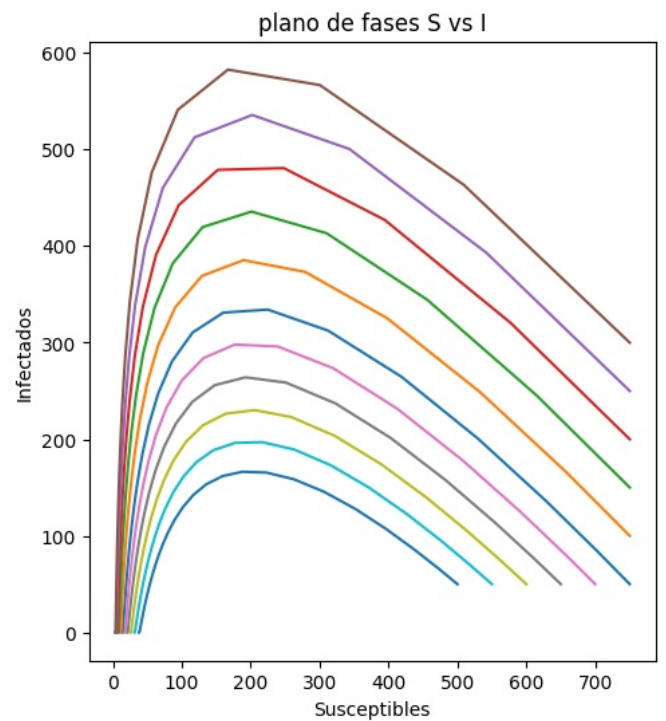
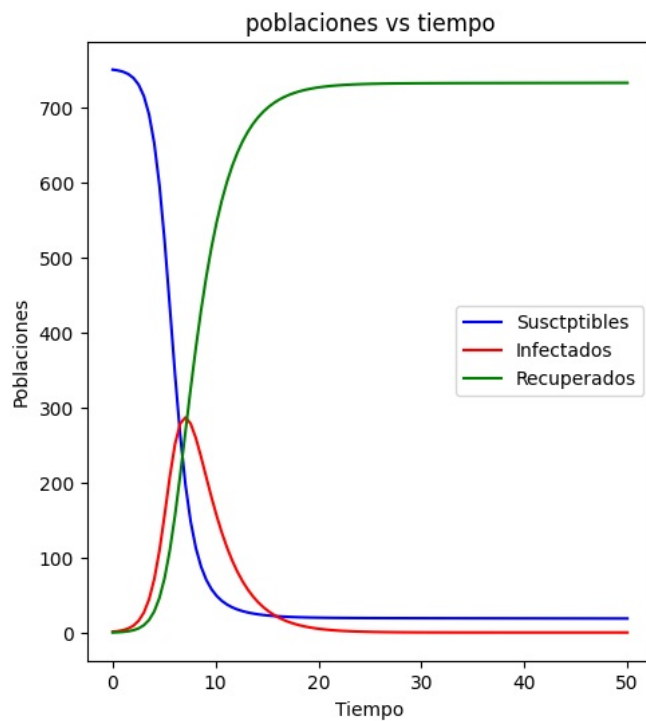
se reperestrna gráficamente la evolución de las poblaciones con respecto al tiempo

```
In [1]: import matplotlib.pyplot as plt
import scipy.integrate as scint
import sympy as sp
import numpy as np
sp.init_printing()
```

```
In [2]: beta=0.002;nu=0.4; xi=0.02
def F(P, t):
    return [-beta*P[0]*P[1] - xi, beta*P[0]*P[1]-nu*P[1], nu*P[1] + xi]
tiempo = np.linspace(0, 50,100)
P0 = [750, 1, 0]
Ps = scint.odeint(F, P0, tiempo)

fig=plt.figure(figsize=(12,6))
plt.subplot(121, title='poblaciones vs tiempo')
plt.plot(tiempo, Ps[:,0], color="blue",label="Suscptibles")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados")
plt.legend()
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")

plt.subplot(122, title='plano de fases S vs I')
for j in range (1,7):
    P02 = [750, j*50, 0]
    Ps2 = scint.odeint(F, P02, tiempo)
    plt.plot(Ps2[:,0],Ps2[:,1])
for j in range (0,5):
    P02 = [700-j*50, 50, 0]
    Ps2 = scint.odeint(F, P02, tiempo)
    plt.plot(Ps2[:,0],Ps2[:,1])
    plt.xlabel("Susceptibles")
    plt.ylabel("Infectados")
plt.xlabel("Susceptibles")
plt.ylabel("Infectados")
plt.show()
```



El modelo SIRS viene dado por el sistema EDO de 2 ecuaciones, necesario para plano de fases y analisis cualitativo:

$$\begin{cases} \frac{dS}{dt} = -\beta SI - \zeta = f_{s,i} \\ \frac{dI}{dt} = \beta SI - \nu I = g_{s,i} \end{cases}$$

## Estudio cualitativo del sistema

```
In [3]: s, i, r = sp.symbols("S I R")
b, x, n, N = sp.symbols("beta xi nu N", positive=True)
fsi = -b*s*i - x*s
gsi = b*s*i - n*i
r = n - s - i
fsi, gsi
```

```
Out[3]: (-I\beta - S\xi, I\beta - I\nu)
```

```
In [4]: p_cr = sp.solve([fsi, gsi], [s, i], simplify=True)
p_cr
```

```
Out[4]: [(0, 0), (N*\nu / (b - x), -N*\xi / (b - x))]
```

Solo sale un punto crítico, pero no presenta ningún sentido biológico, ya que los infectados toman valor negativo.

```
In [5]: par_matrix = sp.Matrix([[fsi.diff(s), fsi.diff(i)], [gsi.diff(s), gsi.diff(i)]])
par_matrix
```

```
Out[5]: [ [-I\beta - \xi   -S\beta ]
[  I\beta      S\beta - \nu ]]
```

## Estabilidad del $P_0$ de equilibrio

```
In [6]: C1 = par_matrix.subs({s: p_cr[0][0], i: p_cr[0][1]})
C1
```

```
Out[6]: [ [-\xi   0 ]
[  0   -\nu ]]
```

```
In [7]: Clauto = C1.eigenvals()
Clautovect = C1.eigenvecs()
Clauto, Clautovect
```



```
Out[7]: 
$$\left( \{-v:1, -\zeta:1\}, \left[ \left( -v, 1, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right), \left( -\zeta, 1, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \right] \right)$$

```

```
In [8]: trazaC1=sp.simplify(sp.trace(C1))
detC1=sp.simplify(sp.det(C1))
trazaC1, detC1
```

```
Out[8]:  $(-v-\zeta, v\zeta)$ 
```

Condición de estabilidad asintótica del sistema lineal:  $\text{tr}(A) < 0$  ;  $\text{det}(A) > 0$

Esto es:

$$\begin{cases} -v-\zeta < 0 \\ v\zeta > 0 \end{cases}$$

Sabiendo que los parámetros del modelo ( $\gamma, v, \beta$  y  $N$ ) vemos que estas condiciones siempre se cumplen. La traza es siempre negativa y el determinante siempre positivo, por lo que el punto es siempre estable.

## Estabilidad de $P_1$ de equilibrio

```
In [9]: C2=par_matrix.subs({s:p_cr[1][0],i:p_cr[1][1]})
C2
```

```
Out[9]: 
$$\begin{bmatrix} 0 & -v \\ -\zeta & 0 \end{bmatrix}$$

```

```
In [10]: trazaC2=sp.simplify(sp.trace(C2))
detC2=sp.simplify(sp.det(C2))
trazaC2, detC2
```

```
Out[10]:  $(0, -v\zeta)$ 
```

En este caso, el valor de la traza siempre es nulo, y el determinante negativo, por lo cual se trata de un punto inestable (punto de silla)

## Comparación con modelos SIR Y SIRS

Vamos a comprobar gráficamente cómo afecta la política de vacunación a las dinámicas de los modelos SIR y SIRS

### SIR vs SIRV y tasa de vacunación independiente de S

Asignamos valores constantes a los parámetros, incluida la tasa de vacunación (constante):

```
In [27]: beta=0.002;nu=0.2;N=750
def F(P, t):
    return [-beta*P[0]*P[1] - xi*P[0], beta*P[0]*P[1]-nu*P[1], nu*P[1] + xi*P[0]]
def G(P, t):
    return [-beta*P[0]*P[1], beta*P[0]*P[1]-nu*P[1], nu*P[1]]
def H(P, t):
    return [-beta*P[0]*P[1] - xi, beta*P[0]*P[1]-nu*P[1], nu*P[1] + xi] #xi es independiente de S

tiempo = np.linspace(0, 50,100)
P0 = [749, 1, 0]

fig=plt.figure(figsize=(12,4))

Pr = scint.odeint(G, P0, tiempo)
plt.subplot(131, title='Modelo SIR')
plt.plot(tiempo,Pr[:,0], color="blue",label="Susceptibles")
plt.plot(tiempo,Pr[:,1], color="red",label="Infectados")
plt.plot(tiempo,Pr[:,2],color="green",label="Recuperados")
plt.plot(tiempo,np.ones(len(tiempo))*max(Pr[:,1]),color='red',linestyle='--')
plt.legend(loc='upper right')
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")

xi=0.1; Ps = scint.odeint(F, P0, tiempo)
plt.subplot(132, title=r'Modelo SIRV ($V(t)=\xi S$). $\xi=\{ \}$.format(xi))
plt.plot(tiempo, Ps[:,0], color="blue",label="Susceptibles")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados")
plt.plot(tiempo,np.ones(len(tiempo))*max(Ps[:,1]),color='red',linestyle='--')
plt.legend(loc='upper right')
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")
```

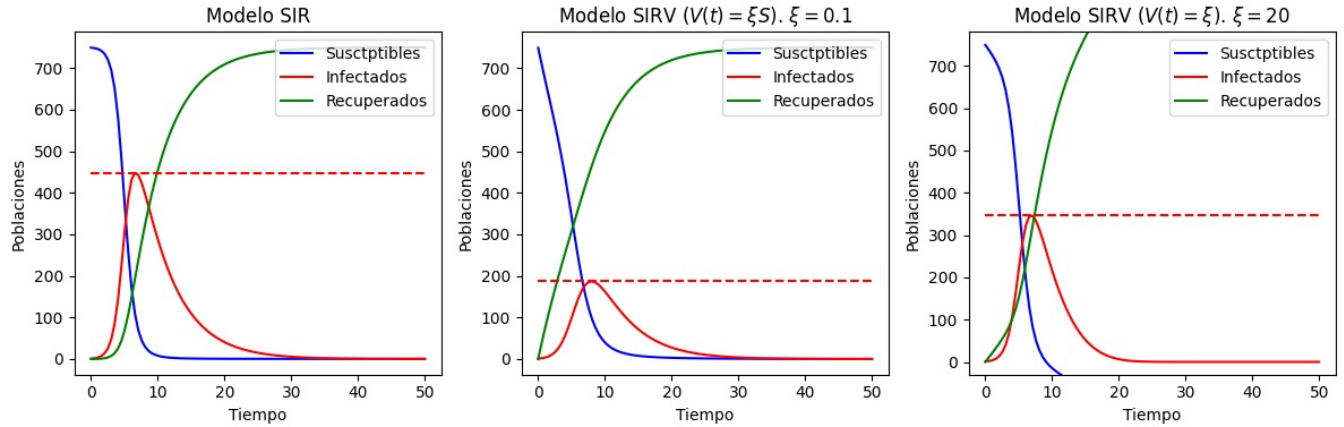


```

xi=20; Pt = scint.odeint(H, P0, tiempo)
plt.subplot(133, title=r'Modelo SIRV ($V(t)=\xi I$). $\xi=\{ \xi \}$.format(xi))
plt.plot(tiempo, Pt[:,0], color="blue",label="Susctptibles")
plt.plot(tiempo,Pt[:,1], color="red",label="Infectados")
plt.plot(tiempo,Pt[:,2],color="green",label="Recuperados")
plt.plot(tiempo,np.ones(len(tiempo))*max(Pt[:,1]),color='red',linestyle='--')
plt.legend()
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")
plt.ylim(-30,780)

plt.savefig('SIR_vs_SIRV.png')
fig.tight_layout()
plt.show()

```



## SIRS vs SIRS

En este caso, el modelo SIRS se rige por la siguiente expresión:

$$\begin{cases} \frac{dS}{dt} = -\beta SI + \gamma R - \xi S \\ \frac{dI}{dt} = \beta SI - \nu I \\ \frac{dR}{dt} = \nu I - \gamma R + \xi S \end{cases}$$

```

In [12]: beta=0.002;nu=0.2; gamma=0.02; xi=0.1
def F(P, t):
    return [-beta*P[0]*P[1] + gamma*P[2] - xi*P[0], beta*P[0]*P[1]-nu*P[1], nu*P[1] - gamma*P[2] + xi*P[0]]
def G(P, t):
    return [-beta*P[0]*P[1] + gamma*P[2], beta*P[0]*P[1]-nu*P[1], nu*P[1] - gamma*P[2]]

tiempo = np.linspace(0, 50,100)
P0 = [750, 1, 0]
Ps = scint.odeint(F, P0, tiempo)
Pr = scint.odeint(G, P0, tiempo)

fig=plt.figure(figsize=(12,3))
plt.subplot(121, title='poblaciones vs tiempo, modelo SIRS')
plt.plot(tiempo,Pr[:,0], color="blue",label="Susctptibles")
plt.plot(tiempo,Pr[:,1], color="red",label="Infectados")
plt.plot(tiempo,Pr[:,2],color="green",label="Recuperados")
plt.legend()
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")

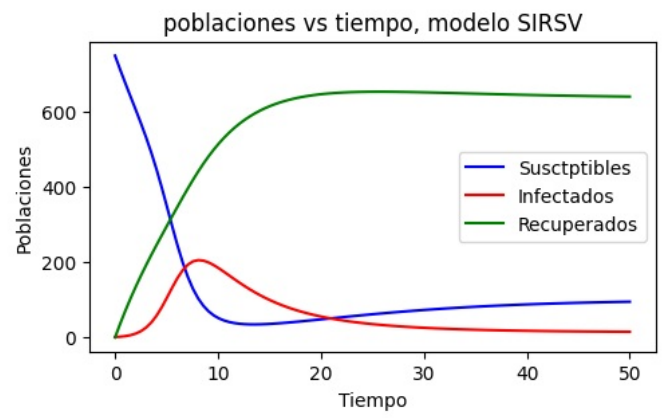
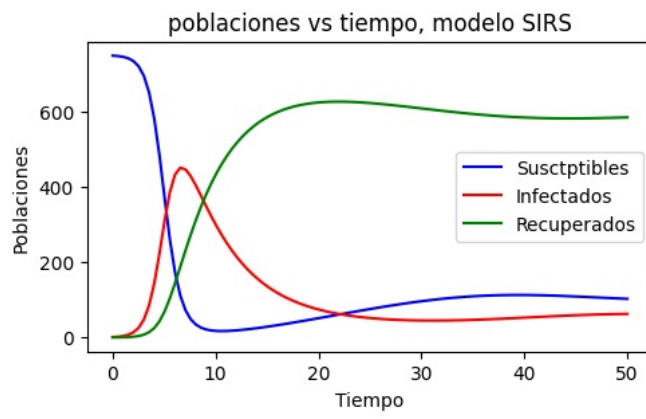
plt.subplot(122, title='poblaciones vs tiempo, modelo SIRS')
plt.plot(tiempo, Ps[:,0], color="blue",label="Susctptibles")
plt.plot(tiempo,Ps[:,1], color="red",label="Infectados")
plt.plot(tiempo,Ps[:,2],color="green",label="Recuperados")
plt.legend()

plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")

plt.savefig('SIRS_vs_SIRS.png')

plt.show()

```



Processing math: 100%

## Modelo epidemiológico SEIR para el covid-19 con medidas de contención

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scint
import sympy as sp
sp.init_printing()
```

Partimos de una población  $N = 100000$ , de los cuales 1000 se encuentran infectados inicialmente, y 99000 sanos. Los valores de los parámetros son los comentados en el artículo.

Gráfica de evolución de poblaciones con el tiempo:

```
In [2]: N, b, s, g = 100000, 1, 1/7, 1/5

def F(P, t):
    return [-b*P[0]*P[2]/N, b*P[0]*P[2]/N - s*P[1], s*P[1]-g*P[2], g*P[2]]

tiempo = np.linspace(0, 100, 100)

P0 = [99000, 0, 1000, 0]

Ps = scint.odeint(F, P0, tiempo)
```

```
In [3]: #print (Ps) (Cada columna pertenece a una población, a su evolución con el tiempo)
```

```
In [4]: # Valores obtenidos de la población 'S'
susceptibles = Ps[:,0]

# Valores obtenidos de la población 'E'
expuestos = Ps[:,1]

#Valores obtenidos de la población 'I'
infectados = Ps[:,2]

# Valores obtenidos de la población 'R'
recuperados = Ps[:,3]

plt.figure(figsize = (8,4))

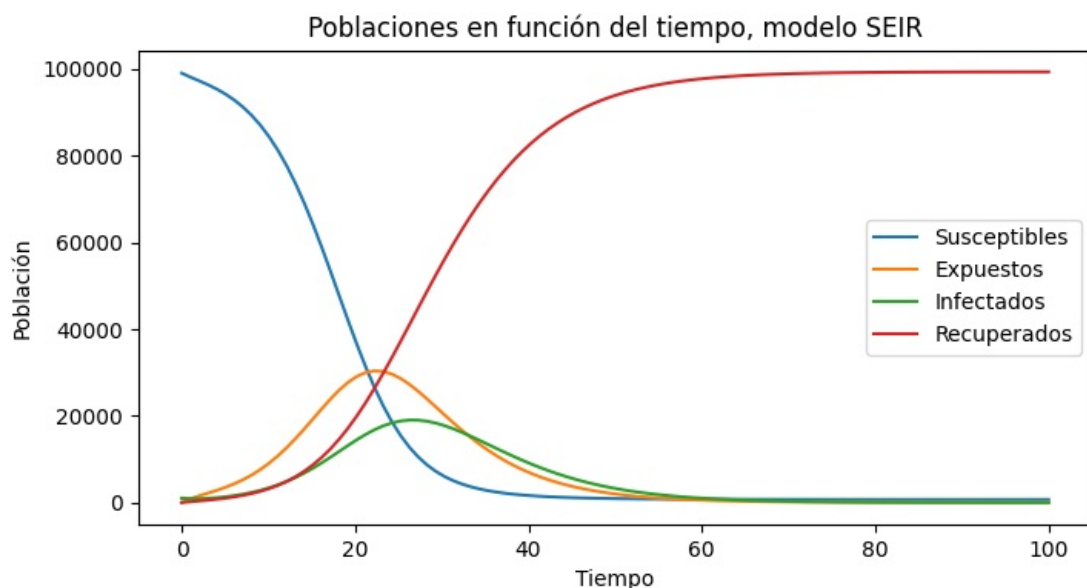
plt.plot(tiempo, susceptibles, label = "Susceptibles")
plt.plot(tiempo, expuestos, label = "Expuestos")
plt.plot(tiempo, infectados, label = "Infectados")
plt.plot(tiempo, recuperados, label = "Recuperados")

plt.xlabel("Tiempo")
plt.ylabel("Población")

plt.title("Poblaciones en función del tiempo, modelo SEIR")
plt.legend()

#plt.savefig('Población_modelo_SEIR.png')

# Mostamos la figura
plt.show()
```



Tasa de transmisión variable con el tiempo

## Tasa de transmisión variable con el tiempo

Incluimos una ecuación más para el parámetro beta. Además, podemos pensar que el 5% de los casos de infección son críticos y conciencian a las personas de la gravedad de la pandemia, con lo que  $D_{(t)} = 0.05I_{(t)}$

```
In [5]: #Valor de los distintos parámetros:
N, b0, s, g, a, k = 100000, 1, 1/7, 1/5, 0.4, 100

def F(P, t):
    return [-(b0*(1-a)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N, (b0*(1-a)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N - s*P[1], s*P[1]]

tiempo = np.linspace(0, 100, 100)

P0 = [99000, 0, 1000, 0]

Pr = scint.odeint(F, P0, tiempo)
```

```
In [6]: #print(Pr)
```

```
In [7]: # Valores obtenidos de la población 'S'
susceptibles = Pr[:,0]

# Valores obtenidos de la población 'E'
expuestos = Pr[:,1]

#Valores obtenidos de la población 'I'
infectados = Pr[:,2]

# Valores obtenidos de la población 'R'
recuperados = Pr[:,3]

# Tamaño de la figura cuadrada
plt.figure(figsize = (8,4))

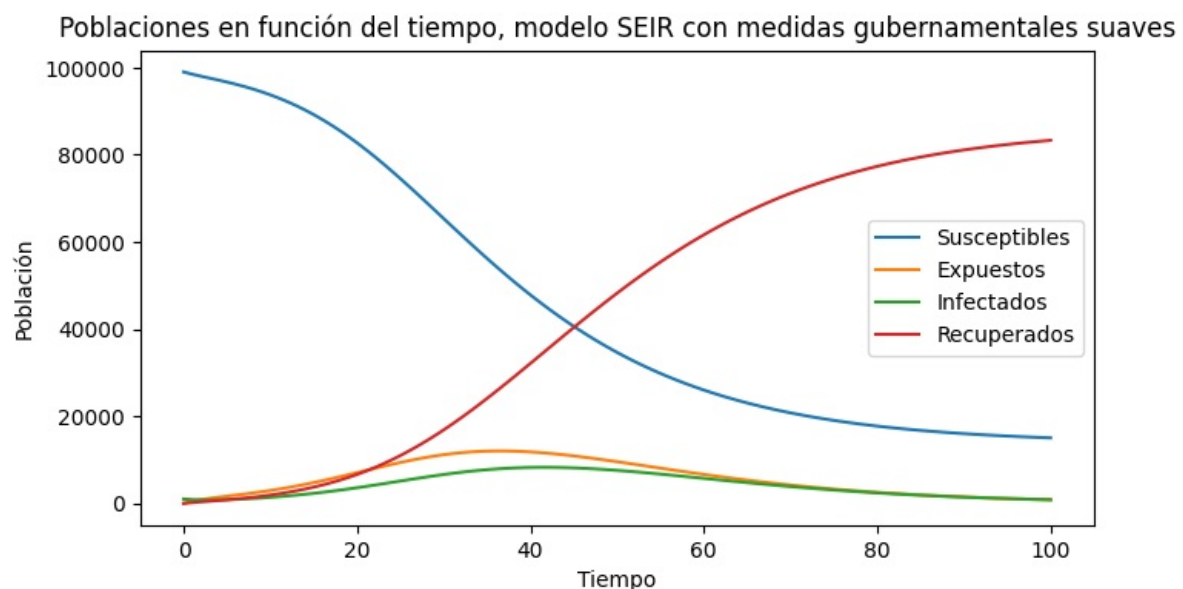
plt.plot(tiempo, susceptibles, label = "Susceptibles")
plt.plot(tiempo, expuestos, label = "Expuestos")
plt.plot(tiempo, infectados, label = "Infectados")
plt.plot(tiempo, recuperados, label = "Recuperados")

plt.xlabel("Tiempo")
plt.ylabel("Población")

plt.title("Poblaciones en función del tiempo, modelo SEIR con medidas gubernamentales suaves")

plt.legend()
plt.savefig('Medidas_suaves.png')

plt.show()
```



Si tenemos medidas de restricción más intensas ( $a = 0.6$ ):

```
In [8]: N, b0, s, g, a, k = 100000, 1, 1/7, 1/5, 0.6, 100

def F(P, t):
    return [-(b0*(1-a)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N, (b0*(1-a)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N - s*P[1], s*P[1]]

tiempo = np.linspace(0, 100, 100)

P0 = [99000, 0, 1000, 0]

Pt = scint.odeint(F, P0, tiempo)
```

```

# Valores obtenidos de la población 'S'
susceptibles = Pt[:,0]

# Valores obtenidos de la población 'E'
expuestos = Pt[:,1]

#Valores obtenidos de la población 'I'
infectados = Pt[:,2]

# Valores obtenidos de la población 'R'
recuperados = Pt[:,3]

# Tamaño de la figura cuadrada
plt.figure(figsize = (8,4))

plt.plot(tiempo, susceptibles, label = "Susceptibles")
plt.plot(tiempo, expuestos, label = "Expuestos")
plt.plot(tiempo, infectados, label = "Infectados")
plt.plot(tiempo, recuperados, label = "Recuperados")

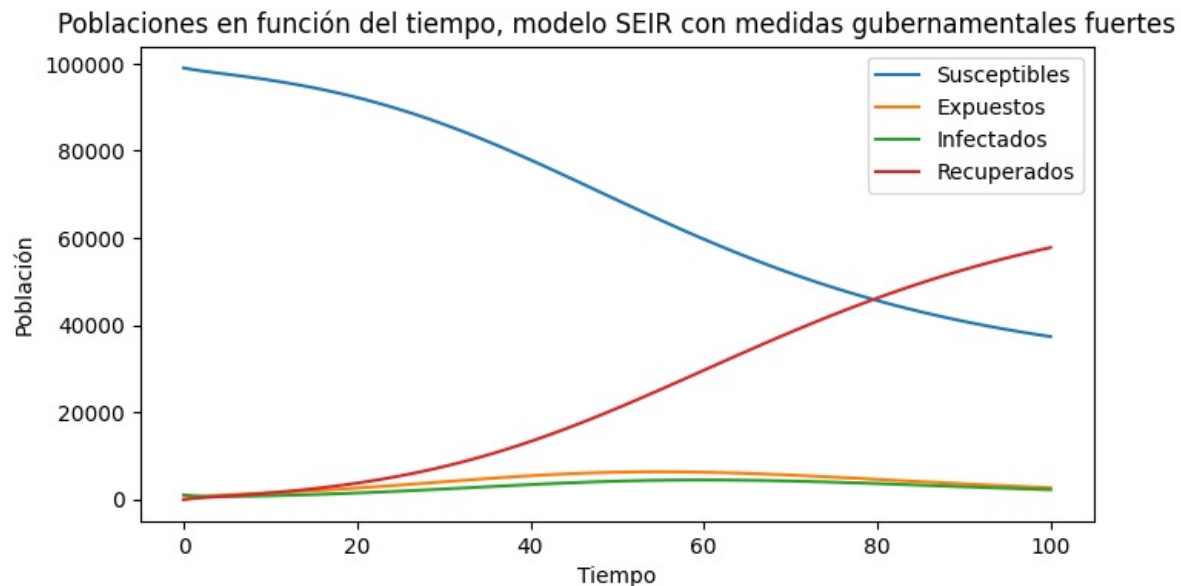
plt.xlabel("Tiempo")
plt.ylabel("Población")

plt.title("Poblaciones en función del tiempo, modelo SEIR con medidas gubernamentales fuertes")

plt.legend()

#plt.savefig('Medidas_fuertes.png')
plt.show()

```



La población de expuestos e infectados aumenta en mucha menor cantidad a medida que implementamos medidas más restrictivas. Los tiempos se prolongan, tarda más en crecer

In [ ]: ## Gráfica conjunta

```

In [9]: N, b0, s, g, k = 100000, 1, 1/7, 1/5, 100
def F(P, t):
    return [-(b0*(1-0.4)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N, (b0*(1-0.4)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N - s*P[1],
def G(P, t):
    return [-(b0*(1-0.6)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N, (b0*(1-0.6)*(1-0.05*P[2]/N)**k)*P[0]*P[2]/N - s*P[1],

tiempo = np.linspace(0, 100,100)
P0 = [99000, 0, 1000, 0]
Ps = scint.odeint(F, P0, tiempo)
Pr = scint.odeint(G, P0, tiempo)

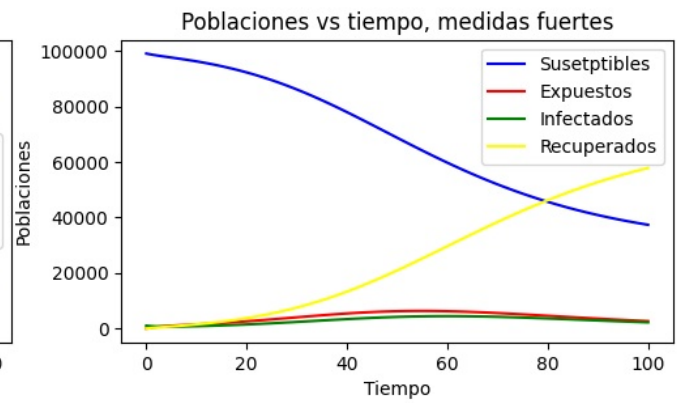
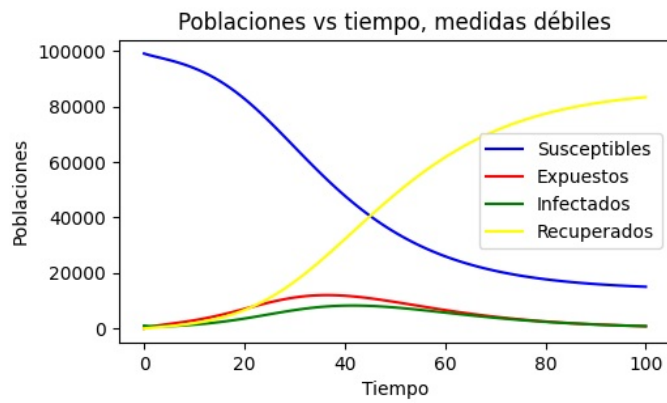
fig=plt.figure(figsize=(12,3))
plt.subplot(121, title='Poblaciones vs tiempo, medidas débiles')
plt.plot(tiempo,Ps[:,0], color="blue",label="Susceptibles")
plt.plot(tiempo,Ps[:,1], color="red",label="Expuestos")
plt.plot(tiempo,Ps[:,2],color="green",label="Infectados")
plt.plot(tiempo,Ps[:,3],color="yellow",label="Recuperados")
plt.legend()
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")

plt.subplot(122, title='Poblaciones vs tiempo, medidas fuertes')
plt.plot(tiempo,Pr[:,0], color="blue",label="Susceptibles")
plt.plot(tiempo,Pr[:,1], color="red",label="Expuestos")
plt.plot(tiempo,Pr[:,2],color="green",label="Infectados")
plt.plot(tiempo,Pr[:,3],color="yellow",label="Recuperados")

```

```
plt.legend()
plt.xlabel("Tiempo")
plt.ylabel("Poblaciones")
```

Out[9]: Text(0, 0.5, 'Poblaciones')



Created in **Deeptime**

Processing math: 100%

# Simulación evolución de la epidemia de ébola en Africa Occidental en 2014

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scint
import pandas as pd
```

Importar datos reales de excel

```
In [2]: df=pd.read_excel("previous-case-counts.xlsx", usecols="J,H")
casos=df['Total Cases'].tolist()[::-1]
dias=df['Days'].tolist()[::-1]
```

Solución numérica de los modelos

```
In [3]: beta=0.0003589
nu=0.1
gamma=0.0179
def F(P, t):
    return [-beta*P[0]*P[1], beta*P[0]*P[1]-nu*P[1], nu*P[1]]

def F2(P, t):
    return [-beta*P[0]*P[1]+gamma*P[2], beta*P[0]*P[1]-nu*P[1], nu*P[1]-gamma*P[2]]

tiempo = np.linspace(0, 700)
P0=[22e6, 49, 0]

Ps=scint.odeint(F, P0, tiempo)
Ps2=scint.odeint(F2, P0, tiempo)
```

```
In [4]: fig, axs = plt.subplots(1, 2, figsize=(15, 4))

axs[0].plot(tiempo, Ps[:,1], color="red", label="Modelo SIR")
axs[1].plot(dias, casos, color="blue")
axs[0].plot(tiempo, Ps2[:,1], color="green", label="Modelo SIRS")
axs[0].set_xlabel("Tiempo (días)")
axs[1].set_xlabel("Tiempo (días)")
axs[0].legend()
axs[0].set_ylabel("Número de casos")
axs[1].set_ylabel("Número de casos")

axs[0].set_title("Evolución epidemia Ébola según modelo SIR")
axs[1].set_title("Evolución epidemia Ébola (datos reales)")

plt.show()
```

