

# Estudio de la pandemia causada por SARS-CoV-2 (2020) mediante modelos epidemiológicos compartimentales

Paula Alóndiga, Pablo Marcos, Víctor Patón

22 de mayo de 2020

## Resumen

La pandemia de SARS-CoV-2, (causante de la enfermedad COVID-19) tiene un alcance global, habiendo infectado ya a más de 4 millones de personas. Para hacer frente a esta crisis, gobiernos de todo el mundo han impuesto medidas más o menos restrictivas para limitar el contagio de la enfermedad, como es el caso de China, la mayoría de la Unión Europea y algunos estados de EEUU. En base a los datos epidemiológicos obtenidos, se han realizado multitud de aproximaciones matemáticas para intentar predecir la evolución de la pandemia.

En este informe se detallan las bases del modelo SIRS y sus sucesivas modificaciones hasta llegar a un modelo ajustado a la evolución de la pandemia. Para ello, se usaron datos de muertes en España entre el 31/12/2019 y el 10/05/2020 proporcionados por la Unión Europea. Además, se evaluaron las medidas de confinamiento decretadas por el Gobierno de España en base al impacto que tienen sobre el número básico de reproducción o  $R_0$ , y el papel de los sistemas sanitarios en la disminución de la mortalidad en base al número de camas UCI disponibles. Con sus limitaciones, creemos que el modelo desarrollado puede predecir con una exactitud aceptable los comportamientos futuros de la pandemia.

Los resultados muestran que las medidas adoptadas han logrado su objetivo, estimándose las vidas salvadas en 1350000 comparado con el peor escenario analizado. Sin embargo, la baja tasa de inmunidad en la población, estimada en 4 % (muy lejana del 60-80 % necesario para la inmunidad de rebaño), hace necesaria la extensión de las medidas de distanciamiento social aún por varios meses.

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Estudio cualitativo del modelo SIRS</b>	<b>3</b>
2.1. Descripción teórica . . . . .	3
2.2. Evaluación de la estabilidad de los puntos críticos . . . . .	4
2.3. Comportamiento global del sistema . . . . .	5
2.4. Limitaciones del modelo . . . . .	6
<b>3. Modelo SEIR modificado para modelado de pandemia por SARS-CoV-2</b>	<b>6</b>
3.1. Descripción teórica . . . . .	6
3.1.1. Impacto del distanciamiento social y/o confinamiento sobre $R_0$ . . .	8
3.1.2. Introducción de las clases Hospitalizados y Críticos . . . . .	8
3.2. Limitaciones del modelo . . . . .	9

<b>4. Evaluación de la efectividad del confinamiento decretado por el Gobierno de España</b>	<b>10</b>
4.1. Confinamiento efectivo el 17/03/2020 . . . . .	10
4.2. Confinamiento efectivo el 10/03/2020 . . . . .	11
4.3. Confinamiento efectivo el 24/03/2020 . . . . .	11
4.4. Sin Confinamiento . . . . .	12
<b>5. Conclusiones</b>	<b>13</b>
<b>6. Métodos</b>	<b>13</b>
<b>Apéndice A. Licencia</b>	<b>14</b>
<b>Apéndice B. Visión global de la epidemia</b>	<b>14</b>
B.1. Lámina 1: Confinamiento efectivo el 17/03/2020 . . . . .	15
B.2. Lámina 2: confinamiento efectivo el 10/03/2020 . . . . .	16
B.3. Lámina 3: confinamiento efectivo el 24/03/2020 . . . . .	17
B.4. Lámina 4: Sin confinamiento . . . . .	18
<b>Apéndice C. Métodos Matemáticos</b>	<b>19</b>
C.1. Modelo SIRS . . . . .	19
C.2. Modelo SEIR Modificado . . . . .	28

## 1. Introducción

El SARS-CoV-2 es un virus de ARN monocatenario positivo, perteneciente a la familia de los Coronavirus y detectado por primera vez en diciembre de 2019 en un paciente diagnosticado por neumonía en Wuhan, China (Kamikubo y Takahashi 2020). Esta familia de virus, que fue catalogada en 2015 por la OMS como "Enfermedad Prioritaria" y con necesidad de investigación urgente por su potencial emergente (OMS 2020), fue la responsable de las epidemias de SARS en 2003 y MERS en 2012, que se caracterizan por provocar enfermedades respiratorias de distinta gravedad y con una mortalidad relativamente alta (WHO 2020).

En el contexto de pandemia global por SARS-CoV-2 en la que se encuentra el planeta, muchos gobiernos tanto en Europa como en el resto de regiones se han visto obligados a adoptar medidas de distanciamiento social, e incluso cuarentenas prolongadas, con el objetivo de disminuir la tasa de nuevos contagios (Martín-Corral y col. 2020). Para medir la efectividad de estas medidas, se ha recurrido a los conocidos como "modelos compartimentales", que se basan en asignar a la población a una serie de compartimentos etiquetados por los cuales va circulando según progresa la enfermedad. Su desarrollo comenzó a principios del Siglo XX, y tuvo importantes aportaciones de Kermack and McKendrick en 1927, quienes sentaron las bases de modelos como SIR, SIRS o SEIR, de gran utilidad hoy en día (Isea y E. Lonngrén 2013).

Estos modelos establecen, como mínimo, los siguientes compartimentos:

- **S:** susceptibles de contraer la enfermedad
- **I:** infectados, contagian la enfermedad

- **R**: han superado la enfermedad, no contagian el virus y han desarrollando inmunidad.

Para lograr un mejor ajuste, y debido a que una estimación adecuada del número de infectados es esencial para evitar el desbordamiento de los servicios sanitarios (Huang y col. 2020 estima que en torno al 30 % de los infectados necesitarán cuidados intensivos), se ha ajustado el modelo mediante el número de muertes, pues son más fáciles de contabilizar dada la escasez de pruebas diagnósticas y el alto número de infectados.

Además, es de especial interés el parámetro  $R_0$ , que denota el número de casos secundarios derivados de un caso de infección (esto es, el número de personas a las que un infectado transmite el virus). Así, cuando este número es inferior a 1, la epidemia decrece con el tiempo, al verse mermada la capacidad del virus de encontrar nuevos huéspedes susceptibles de contagio; mientras que la misma puede crecer exponencialmente si este número se dispara (Liu y col. 2018).

Sin embargo, estos modelos, aunque tienen grandes virtudes, también tienen evidentes limitaciones derivadas de su sencillez. Por ello se ha procedido a modificar el modelo SEIR para obtener un mejor marco teórico sobre el que realizar las predicciones concernientes a la situación general y el efecto de las medidas de confinamiento en España.

## 2. Estudio cualitativo del modelo SIRS

### 2.1. Descripción teórica

Como hemos explicado, se trata de un modelo determinista basado en distintos compartimentos por los que la población va transitando según se expone al virus: Susceptibles, Infectados y Recuperados. Este modelo asume que no se producen ni nacimientos ni muertes, y viene definido por las siguientes ecuaciones:

$$\frac{dS}{dt} = -\beta IS + \nu R \quad (1)$$

$$\frac{dI}{dt} = \beta IS - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma I - \nu R \quad (3)$$

Asumiendo que la población total es constante, podemos eliminar  $R$  de las anteriores sustituyendo  $R = N - S - I$ , de donde obtendríamos:

$$\frac{dS}{dt} = -IS\beta + \nu(-I + N - S) \quad (4)$$

$$\frac{dI}{dt} = IS\beta - I\gamma \quad (5)$$

Esta ecuación presenta los parámetros que se detallan a continuación:

$\beta$  : Número de personas infectadas por individuo enfermo y día.

$\gamma$  : Inversa del periodo de infección. Es la tasa a la que se recuperan los infectados.

$\nu$  : Tasa de pérdida de inmunidad

## 2.2. Evaluación de la estabilidad de los puntos críticos

Para evaluar los puntos críticos, hacemos uso del módulo numpy de python. Tras resolver y simplificar, obtenemos los puntos:

$$(N, 0) \quad \left( \frac{\gamma}{\beta}, \frac{\nu(N\beta - \gamma)}{\beta(\gamma + \nu)} \right)$$

De estos dos puntos,  $(N, 0)$  se establece cuando la población está sana y el virus ausente, mientras que el otro, al que llamaremos  $(S_2^*, I_2^*)$ , describe el equilibrio al que tiende la población de individuos susceptibles y sanos. Es este segundo punto el que nos resulta de mayor interés. Así, procedemos a calcular el jacobiano, de forma que:

$$\begin{bmatrix} -\nu & N\beta - \nu \\ 0 & N\beta - \gamma \end{bmatrix}$$

Además, podemos obtener tanto la traza como el determinante de la matriz de interacción para  $(S_2^*, I_2^*)$ :

$$\text{Traza} : \nu(N\beta - \gamma) \quad \text{Determinante} : \frac{-\nu(N\beta + \nu)}{\gamma + \nu}$$

De los que se deduce que existen condiciones de estabilidad para los cuales los autovalores son complejos. Si ahora procedemos a dibujar las órbitas que se producen cerca de los puntos de equilibrio, podemos ver que:

- Para el punto  $(N, 0)$ , tenemos un punto de silla fuente, es decir, altamente inestable: en cuanto aparecen algunos casos, se tiende al segundo punto.
- Para el punto  $(S_2^*, I_2^*)$ , tenemos un nodo o espiral sumidero, por lo que todos los valores tienden a este punto de equilibrio una vez se ha sembrado la infección.

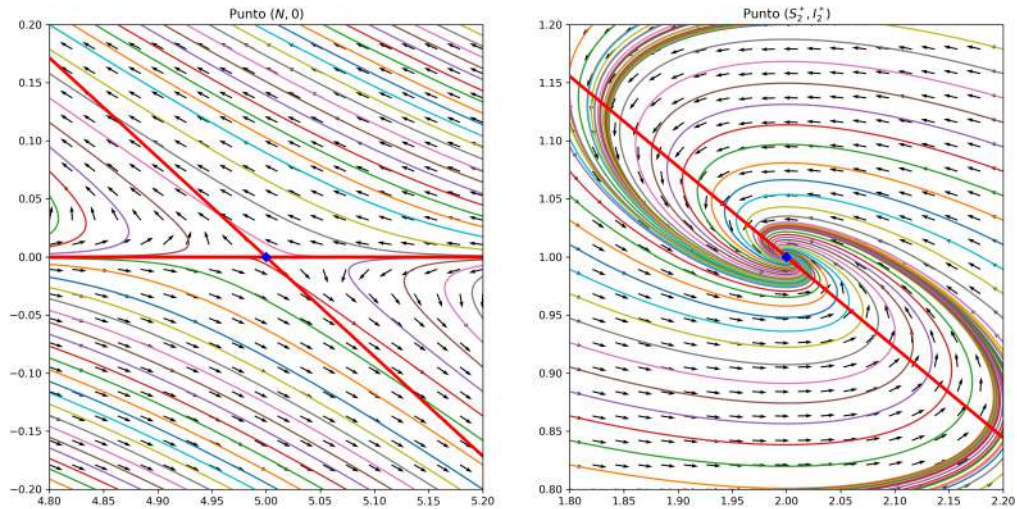


Figura 1: Puntos de equilibrio del Modelo SIRS para unos valores iniciales de los parámetros de  $\beta = 1$ ,  $\nu = 2$ ,  $\gamma = 1$ ,  $N = 5$

### 2.3. Comportamiento global del sistema

Si nos fijamos en la evolución global del sistema, podemos ver que lo establecido antes se cumple, siendo el punto  $(S_2^*, I_2^*)$  el único estable cuando existe población infectada.

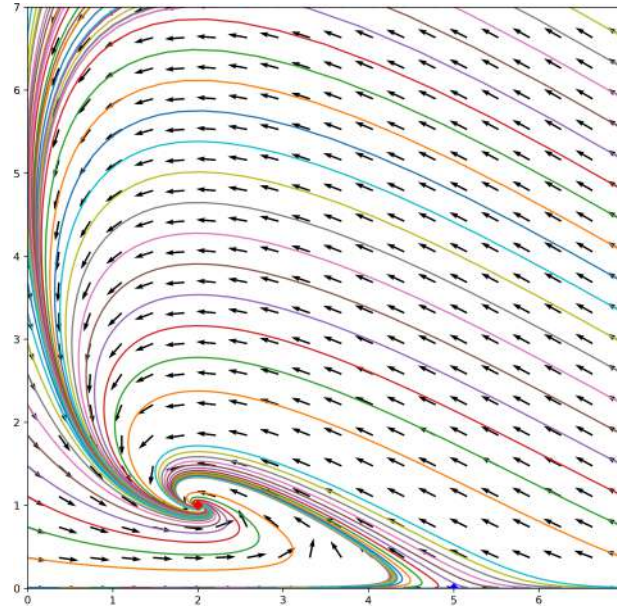


Figura 2: Comportamiento Global del Sistema SIRS para los valores especificados

Así pues, la población sufre una evolución a lo largo del tiempo consistente con los equilibrios especificados, que nos lleva, como se puede ver en la figura a continuación, a un crecimiento rapidísimo del número de infectados, simultáneo al aumento exponencial de los recuperados y, al verse exhaustada la capacidad de infección por las defensas de la población, una bajada igualmente rápida de los susceptibles.

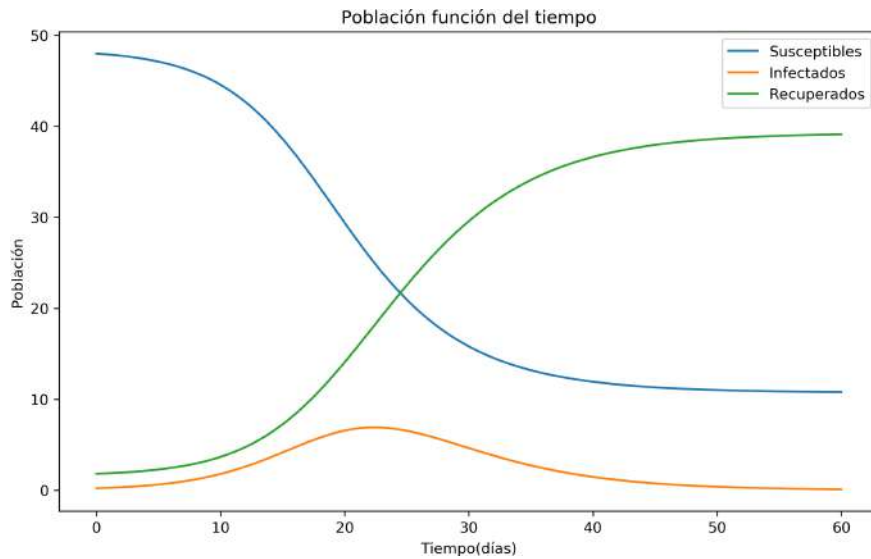


Figura 3: Población en función del tiempo de los distintos compartimentos del modelo SIRS

## 2.4. Limitaciones del modelo

El SIRS es un modelo muy sencillo que puede capturar parte, pero no toda, de la complejidad de una epidemia: aunque permite reinfecciones, asume una población sin nacimientos ni muertes, y define unas capacidades inmunológicas homogéneas en toda la población, algo que sabemos que no es así (McCluskey 2010). Sin embargo, sirve para introducir los conceptos básicos de  $R_0$  y otras calculaciones asociadas, así como otros parámetros que siguen siendo esenciales para la toma en consideración de una enfermedad (Bjørnstad, Finkenstädt y Grenfell 2002 y Weiss 2020).

## 3. Modelo SEIR modificado para modelado de pandemia por SARS-CoV-2

Dadas las limitaciones del modelo SIRS básico estudiado en la sección anterior, es necesaria una aproximación teórica más compleja. El modelo SEIR, que data del siglo XX (Kermack y McKendrick 1927), añade al SIR/SIRS un nuevo compartimento, **E**xpuestos, que define a las personas que han estado en contacto con un infectado, pero no son todavía infecciosos por si mismos; es decir, se asume la existencia de un periodo de latencia no infectiva.

### 3.1. Descripción teórica

Se definen 7 compartimentos entre los que se distribuye la población:

- **S**: Susceptibles de contraer la enfermedad
- **E**: Expuestos a infectados, incuban la enfermedad pero aún no son contagiosos.
- **I**: Infectados, contagian la enfermedad
- **R**: Han superado la enfermedad, no contagian el virus y poseen inmunidad.
- **H**: Individuos infectados que requieren ingreso hospitalario.
- **C**: Individuos infectados que requieren cuidados intensivos.
- **M**: Fallecidos por COVID-19

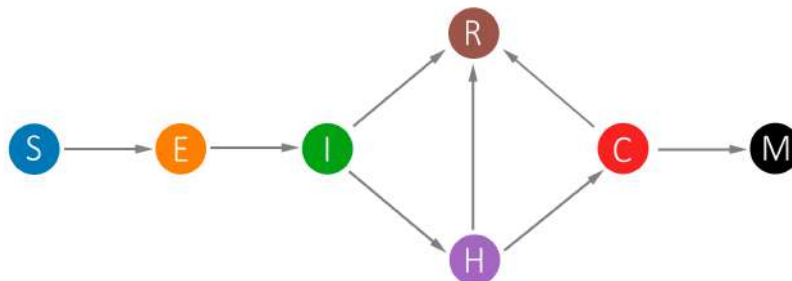


Figura 4: Compartimentos del modelo y los flujos que se establecen entre ellos

Cada uno de los flujos viene determinado por 3 componentes: la probabilidad de que un individuo cambie de un compartimento a otro, la tasa a la que lo hace (1 / número de días que tarda) y el grupo al que pertenece dicho individuo. Esto, aplicado a los flujos descritos en la Figura 4, permite un sistema de ecuaciones diferenciales ordinarias a partir del cual desarrollaremos un modelo de predicción:

$$\frac{dS}{dt} = -\beta I \frac{S}{N} \quad (6)$$

$$\frac{dE}{dt} = \beta I \frac{S}{N} - \delta E \quad (7)$$

$$\frac{dI}{dt} = \delta E - (1 - a) \gamma I - \eta a I \quad (8)$$

$$\frac{dH}{dt} = \eta (1 - a) I - \tau (1 - u) H - u \sigma H \quad (9)$$

$$\frac{dC}{dt} = u \sigma H - m \rho \min(UCI, C) - \max(0, C - UCI) - \omega (1 - m) \min(UCI, C) \quad (10)$$

$$\frac{dR}{dt} = \gamma (1 - a) I + \tau (1 - u) H + \omega (1 - m) \min(UCI, C) \quad (11)$$

$$\frac{dM}{dt} = m \rho \min(UCI, C) + \max(0, C - UCI) \quad (12)$$

Los parámetros que aparecen en estas ecuaciones se detallan a continuación:

#### Tasas

$\beta$  : Número de personas infectadas por individuo enfermo y día.

$\gamma$  : Inversa del periodo de infección. Es la tasa a la que se recuperan los infectados.

$\delta$  : Inversa del periodo de incubación. La tasa a la que los individuos que incuban el virus desarrollan síntomas y pasan a ser infecciosos.

$\eta$  : Inversa del periodo de desarrollo de síntomas que requieren ingreso hospitalario.

$\tau$  : Inversa del periodo de estancia hospitalaria hasta recuperación en casos que no requieren ingreso en UCI.

$\sigma$  : Inversa del periodo de desarrollo de síntomas que requieren ingreso en UCI. En este modelo, una parte de los infectados no se recuperan, sino que pasan a ser críticos y requieren estancia en UCI

$\rho$  : Inversa del tiempo de estancia en UCI hasta la muerte.

$\omega$  : Inversa del tiempo de estancia en UCI hasta recuperación.

## Probabilidades

- a: Probabilidad de que un individuo infectado necesite ingreso hospitalario
- u: Probabilidad de que un individuo hospitalizado pase a estar crítico.
- m: Probabilidad de que un individuo en estado crítico fallezca.

### 3.1.1. Impacto del distanciamiento social y/o confinamiento sobre $R_0$

El número básico de reproducción  $R_0$  es vital para la evolución de una pandemia. Todas las acciones recomendadas o impuestas desde el Estado van encaminadas a disminuir este parámetro a valores inferiores a 1, pues, como se ha visto, el virus desaparece con el tiempo en este caso. Por ello, la aproximación presentada en este estudio es simple pero efectiva: se teoriza que, a partir de una fecha  $\mathbf{A}$ , en la cual entran en vigor las medidas de confinamiento a gran escala, la tasa  $R_0$  pasa de un valor alto a uno más bajo.

Sin embargo, la disminución de  $R_0$  no es brusca, sino que la transición dura varios días (antes y después de la fecha  $\mathbf{A}$ ) debido a distintos motivos (instauración de teletrabajo, cierre de instituciones educativas...) que ya de por sí reducen moderadamente la movilidad de los ciudadanos. Para simular este comportamiento, se ha utilizado una función logística:

$$R_0(t) = \frac{R_0i - R_0f}{1 + e^{-k(A-t)}} + R_0f \quad (13)$$

Sin embargo,  $R_0$  como tal no aparece en el modelo propuesto al inicio de esta sección. Este parámetro está relacionado con  $\beta$  y  $\gamma$ , el número de casos secundarios generados por día y la tasa de recuperación de los infectados, respectivamente. Por tanto, la relación entre  $R_0$  y el modelo se establece a través del cálculo de  $\beta(t)$ :

$$R_0(t) = \frac{\beta}{\gamma} \Rightarrow \beta(t) = R_0(t) \cdot \gamma \quad (14)$$

### 3.1.2. Introducción de las clases Hospitalizados y Críticos

En el modelo se describen dos clases adicionales al modelo SEIR básico: **Hospitalizados** y **Críticos**.

En base a la literatura, se calcula que entre un 5 y un 25 % de los infectados desarrollan síntomas que requieren asistencia hospitalaria (Rascado Sedes y Ballesteros Sanz 2020). Esto implica un uso de recursos hospitalarios mayor que los casos leves, y por tanto es necesario cuantificar el estrés adicional al que se someten los recursos hospitalarios a causa de la pandemia. En España se calcula que existen más de 100000 camas hospitalarias, por lo que en principio el sistema debería ser solvente para prestar asistencia a todo el que lo necesite. Por tanto, no tendremos en cuenta situaciones de escasez de camas de cuidado agudo, ya que el verdadero cuello de botella de los sistemas sanitarios se encuentran en las plazas de **cuidados intensivos**.

La clase críticos aglutina a aquellos individuos que han resultado infectados por el virus y que requieren de cuidados intensivos para recuperarse. Es una medida indirecta de la fortaleza de los sistemas de salud y bienestar de un país a la hora de hacer frente a una pandemia de estas características. Resulta intuitivo pensar que un país con una renta alta y, por tanto, con capacidad de invertir en un sistema sanitario robusto, va a tener una



tasa de mortalidad menor que un país de ingresos bajos, a menudo con infraestructuras sanitarias precarias. Para llevar a cabo esta aproximación, tomamos como referencia el número de camas UCI por 100000 habitantes. En España, esta cifra es de 10.3 camas UCI por cada 100000 habitantes según Rascado Sedes y Ballesteros Sanz 2020

Esto implica la existencia de una capacidad máxima a partir de la cual el sistema sanitario se colapsa. Por tanto, se debe asumir que los pacientes que requieren cuidados intensivos pero no los reciben por falta de medios fallecen con una probabilidad del 100 %. Esta es una asunción necesaria para mantener la complejidad del modelo en unos niveles manejables. Cuando la epidemia cobra gravedad, las administraciones empiezan a aumentar la capacidad de los sistemas sanitarios mediante la instalación de hospitales de campaña, hoteles medicalizados o aumento de camas de UCI. Esta capacidad aumenta a una tasa  $s$ , que hemos estimado como 0.012, y que en el modelo desarrollado va a indicar la tasa de aumento de camas de UCI y, por tanto, la capacidad asistencial a los pacientes críticos:

$$Camas\ UCI(t) = Camas\ de\ UCI_0 + s \cdot Camas\ de\ UCI_0 \cdot t \quad (15)$$

En definitiva, con este parámetro se evalúa la capacidad de respuesta de las administraciones frente a una crisis sanitaria como esta.

### 3.2. Limitaciones del modelo

Con todo, el modelo posee ciertas limitaciones, algunas inherentes al mismo y otras debido a los datos con los que se ha construido. La primera limitación es el hecho de que el modelo considera una población  $N$  homogénea y equitativamente distribuida, en lugar de tener en cuenta la localización geográfica de los infectados, ni las redes interpersonales que se establecen entre las personas y comunidades. Esto implica que, por ejemplo, un infectado de Galicia tiene las mismas posibilidades de infectar a una persona de Sevilla que a su vecino, aunque resulta evidente que esto no es así.

Por otro lado, el modelo no contempla fenómenos de desescalada: hemos decidido no incluirlo en el estudio debido a que no existe una serie de datos lo suficientemente significativa para poder extrapolar un comportamiento a partir de ella. Por tanto,  $R_0$  siempre disminuye o permanece constante.

Además, asumimos que una persona recuperada genera inmunidad a largo plazo, pero algunos estudios muestran que tal vez cierto porcentaje de recuperados pierden la inmunidad tras un corto espacio de tiempo. De hecho, se han descrito casos de individuos que dieron negativo a las pruebas PCR pero poco después se reactivó la infección, por lo que no se sabe si existe transmisión entre las clases susceptibles y expuestos.

Otra asunción que se hace es el hecho de que no contempla la cronificación de la enfermedad, esto es, la existencia de individuos que precisan cuidados durante un largo período de tiempo y que por tanto son una carga adicional para el sistema sanitario. En esta línea, la evaluación de las capacidades del sistema sanitario sólo por la tasa de camas de UCI por población es claramente insuficiente para valorar el comportamiento de un sistema tan complejo. Sin embargo, es necesaria esta simplificación en aras de la comodidad a la hora de trabajar con este modelo.

## 4. Evaluación de la efectividad del confinamiento decretado por el Gobierno de España

El Gobierno de España, ante el aumento de casos en el país y la evidencia de que existía ya una transmisión comunitaria descontrolada en amplias áreas del territorio, decidió limitar el derecho a la movilidad bajo el paraguas legal del Estado de Alarma. Esto ha provocado una reducción de los desplazamientos de más de un 90 % (Google 2020) y, por tanto, el desplome del número básico de reproducción a niveles inferiores a 1. Esto ha impedido la progresión exponencial que venía siguiendo el número de casos diarios y ha permitido que la saturación del sistema sanitario no sea catastrófica. Además, se han aumentado los recursos sanitarios como la construcción de hospitales de campaña (IFEMA, por ejemplo), lo que eleva la tasa  $s$ .

Los parámetros calculados a partir del número de muertes son los siguientes ( $\delta$ ,  $\gamma$ ,  $\rho$ ,  $\omega$  y  $\sigma$  ya se conocen de la literatura):

- |                       |                       |                   |                   |
|-----------------------|-----------------------|-------------------|-------------------|
| ▪ $A \simeq 73.00$    | ▪ $R_0 f \simeq 0.84$ | ▪ $u \simeq 0.15$ | ▪ $a \simeq 0.13$ |
| ▪ $R_0 i \simeq 4.24$ | ▪ $k \simeq 5.39$     | ▪ $m \simeq 0.20$ |                   |

En los ensayos que se presentan a continuación se ha realizado una previsión de la evolución de la epidemia a 7 meses vista desde el inicio de la serie, desde el 31/12/2019 hasta el 01/07/2020. Los datos que se comparan son del 01/05/2020.

### 4.1. Confinamiento efectivo el 17/03/2020

En este apartado se evalúa la situación actual. Se calculan un número teórico de 24921 muertes a 01/05/2020, muy cercano al real de 24275, con lo que constatamos que el ajuste a los datos es bueno. Las medidas de confinamiento tienen un impacto notable en la tendencia exponencial de la curva de infectados, pero no son suficientes para impedir la sobrecarga del sistema sanitario. Por ello, se observa un pico de muertes (similar al pico de muertes que se registró a inicios de abril: 03/04/2020 950 muertes) en la Figura 5.

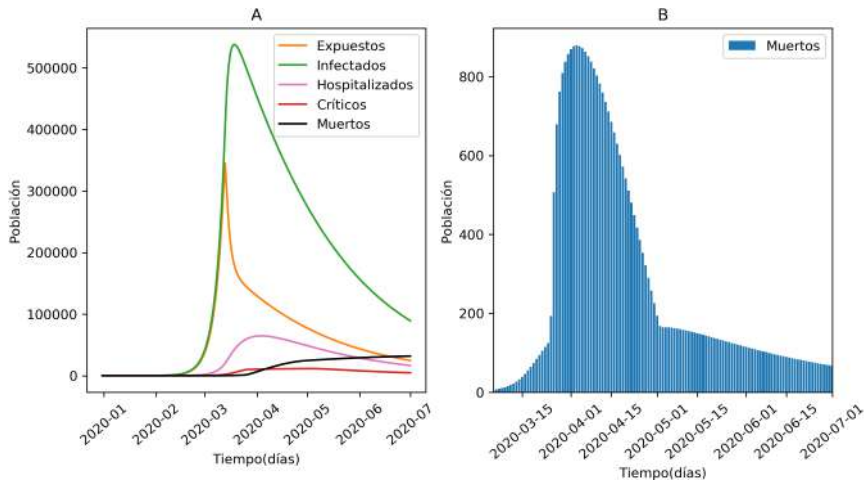


Figura 5: **A:** detalle de la cinética de los compartimentos E, I, H, C y M.

**B:** número diario de fallecidos por COVID-19 al decretarse confinamiento el 17/03/2020

	11Susceptibles	11Expuestos	11Infectados	11Hospitalizados	11Críticos	11Recuperados	11Muertos
Población	44161108	75603	269282	47876	11558	2435855	24921

Cuadro 1: Recuento de población de cada clase, 01/05/2020

En base al número de muertes, se calcula que 2435855 personas han superado la enfermedad y, por tanto, serían inmunes a sucesivas oleadas del virus (Cuadro 1)

#### 4.2. Confinamiento efectivo el 10/03/2020

En este apartado se intenta determinar cual sería el estado actual de la pandemia si el confinamiento se hubiera decretado el 10/03/2020 (una semana antes). Se calculan un número teórico de 3795 muertes a 01/05/2020, un 85 % menos en comparación con el conteo real. Esto nos permite suponer que la enfermedad tuvo una difusión muy elevada entre el 10 y el 17 de marzo (Figura 6).

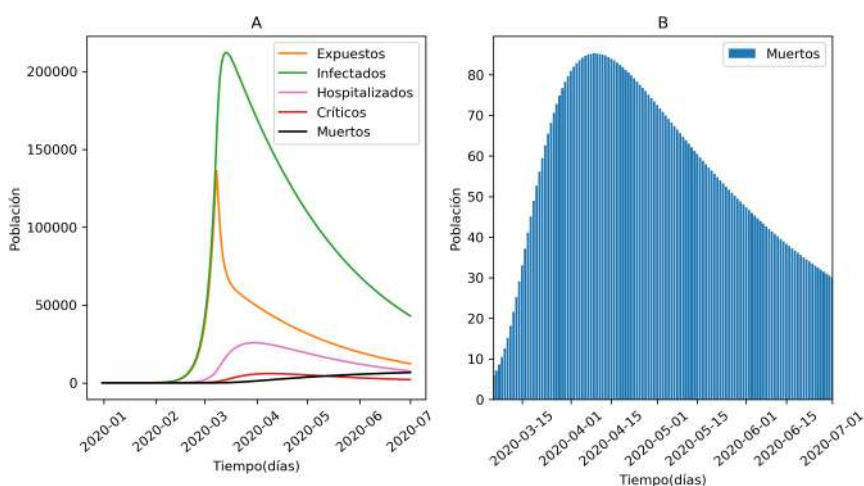


Figura 6: **A:** detalle de la cinética de los compartimentos E, I, H, C y M. **B:** número diario de fallecidos por COVID-19 al decretarse confinamiento el 10/03/2020

	11Susceptibles	11Expuestos	11Infectados	11Hospitalizados	11Críticos	11Recuperados	11Muertos
Población	45806429	31093	107740	18744	4991	1053412	3795

Cuadro 2: Recuento de población de cada clase, 01/05/2020

En base al número de muertes, se calcula que 1053412 personas habrían superado la enfermedad y, por tanto, serían inmunes a sucesivas oleadas del virus (Cuadro 2)

#### 4.3. Confinamiento efectivo el 24/03/2020

En este apartado se intenta determinar el estado actual de la pandemia si el confinamiento se hubiera decretado el 24/03/2020 (una semana después). Se calculan un número teórico de 202565 muertes a 01/05/2020, unas 20 veces más en comparación con el conteo real. Atrasar una semana la restricción de la movilidad habría provocado una pérdida de vidas sin precedentes en tiempos de paz, colapsando el sistema sanitario por completo (Figura 7).

En base al número de muertes, se calcula que 7093729 personas habrían superado la enfermedad y, por tanto, serían inmunes a sucesivas oleadas del virus (Cuadro 3).

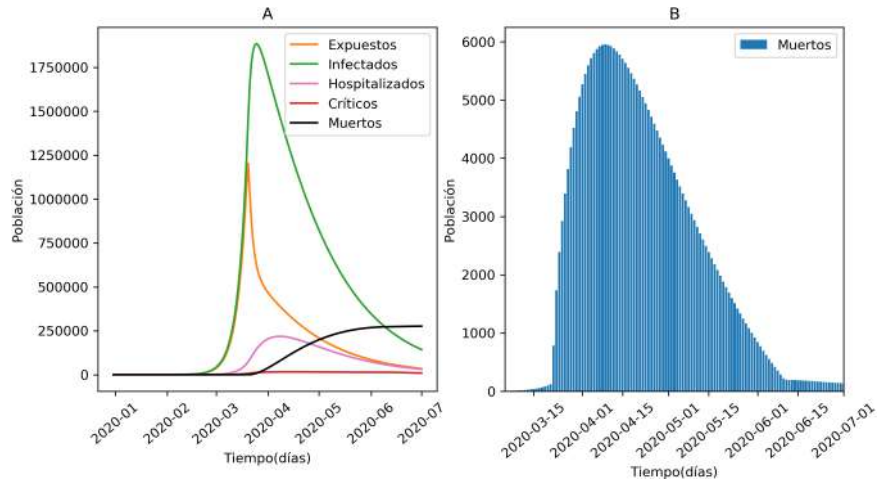


Figura 7: **A:** detalle de la cinética de los compartimentos E, I, H, C y M.  
**B:** número diario de fallecidos por COVID-19 al decretarse confinamiento el 24/03/2020

	1 Susceptibles	1 Expuestos	1 Infectados	1 Hospitalizados	1 Críticos	1 Recuperados	1 Muertos
Población	38552238	203492	803650	155297	15232	7093729	202565

Cuadro 3: Recuento de población de cada clase, 01/05/2020

#### 4.4. Sin Confinamiento

En este apartado se intenta determinar el estado actual de la pandemia si no se hubieran tomado medidas drásticas de reducción de la movilidad. Se calculan un número teórico de 1368282 muertes a 01/05/2020, casi 100 veces más en comparación con las cifras reales. Para entonces, gran parte de la población habría pasado ya la enfermedad y el virus estaría cercano a su desaparición (debido a la inmunidad de rebaño obtenida tras superar la enfermedad grandes masas de población). Sin embargo, las secuelas en vidas humanas serían inasumibles, además de un colapso total del sistema sanitario(Figura 8).

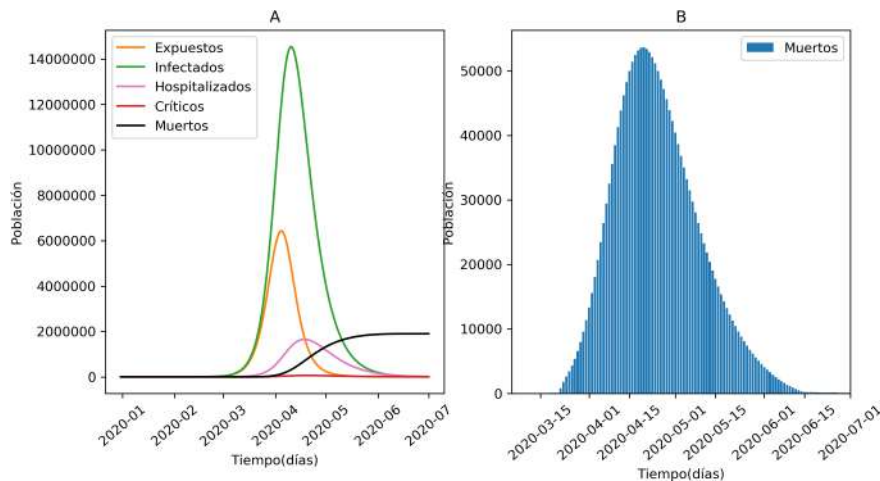


Figura 8: **A:** detalle de la cinética de los compartimentos E, I, H, C y M.  
**B:** número diario de fallecidos por COVID-19 sin confinamiento

En base al número de muertes, se calcula que 39554503 personas habrían superado la enfermedad y, por tanto, son inmunes a sucesivas oleadas del virus (Cuadro 4).

	11Susceptibles	11Expuestos	11Infectados	11Hospitalizados	11Críticos	11Recuperados	11Muertos
Población	1052540	203880	3675727	1122122	49151	39554503	1368282

Cuadro 4: Recuento de población de cada clase, 01/05/2020

## 5. Conclusiones

En esta fase temprana de toma de medidas de contención, y en la que no hay aún posibilidad de vacuna, hemos podido analizar las medidas de control social y su efecto en la reducción de números de contactos, lo que se traduce directamente en vidas salvadas, tanto por los menores contactos como por la posibilidad de tratar adecuadamente a los que han sido diagnosticados.

Se constata así una considerable reducción en la  $R_0$  (de más de tres puntos), lo que representa un éxito al permitir el paso de la fase de crecimiento exponencial a una lenta desescalada, pues los datos de  $R_0$  logran bajar de 1. Nuestras simulaciones concluyen que, de no haberse aplicado el confinamiento, el número de infectados acumulados podría haber alcanzado casi 40 millones, con en torno a 2 millones de muertes; es decir, que con estas medidas, se ha logrado salvar más de 1300000 vidas, evitando además la saturación completa y absoluta del sistema sanitario.

Los autores de este artículo somos conscientes de que los datos de infecciones probablemente presenten una infravaloración de al menos un orden de magnitud, debido no sólo a la falta de equipos de test, sino a la presencia de defectos en muchos de ellos, así como a la no existencia de estudios poblacionales aleatorizados, que probablemente no sean publicados en su versión definitiva hasta Junio. A pesar de ello, la tasa de contagios no es tan alta como cabría esperar, y, consistentemente con literatura previa (London 2020), está lejos del 60-80 % que otros estudios predicen necesario para alcanzar la inmunidad de rebaño, por lo que es posible que las restricciones sociales estén aquí para quedarse. Tal vez las aplicaciones móviles como las desarrolladas por el PEPP-PT o el DP-3T y el contact tracing sean, a futuro y hasta que se desarrolle una vacuna, las medidas más prometedoras a la hora de encontrar una solución que permita la vuelta a la normalidad.

## 6. Métodos

Para realizar este trabajo, primero hemos estudiado un modelo SIRS simple, y después hemos ajustado los datos de muertes sobre un modelo SEIR modificado usando python y Jupyter Notebooks, tomando unas estimaciones previas de los valores frontera de los diferentes parámetros (Rascado Sedes y Ballesteros Sanz 2020). Nos hemos basado ampliamente en un modelo definido por Froese 2020 en el sitio web "Towards Data Science", y, para la realización del documento, hemos usado el módulo xelatex de latex.

Los cálculos concretos usados en el modelo, así como los archivos de datos y los métodos matemáticos utilizados, son parte del archivo comprimido adjunto al documento, y se han obtenido de la Comisión Europea European Commission 2020 y del INE. En los anexos, se pueden consultar figuras de resultados pormenorizados para las distintas simulaciones, así como PDFs de los cuadernos de Jupyter

## Referencias

- Bjørnstad, Ottar N., Bärbel F. Finkenstädt y Bryan T. Grenfell (2002). “Dynamics of Measles Epidemics: Estimating Scaling of Transmission Rates Using a Time Series Sir Model”. en. En: *Ecological Monographs* 72.2, págs. 169-184. ISSN: 1557-7015. DOI: 10.1890/0012-9615(2002)072[0169:DOMEES]2.0.CO;2. URL: <https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/0012-9615%282002%29072%5B0169%3ADOMEES%5D2.0.CO%3B2> (visitado 17-05-2020).
- European Commission (mayo de 2020). *COVID-19 Coronavirus data - ecodp.common.ckan.site\_title*. en. URL: <https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data> (visitado 20-05-2020).
- Froese, Henri (abr. de 2020). *Infectious Disease Modelling: Fit Your Model to Coronavirus Data*. en. URL: <https://towardsdatascience.com/infectious-disease-modelling-fit-your-model-to-coronavirus-data-2568e672dbc7> (visitado 17-05-2020).
- Google (2020). *COVID-19 Community Mobility Report*. publisher: Google. URL: <https://www.google.com/covid19/mobility?hl=es> (visitado 17-05-2020).
- Huang, Chaolin y col. (feb. de 2020). “Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China”. en. En: *The Lancet* 395.10223, págs. 497-506. ISSN: 01406736. DOI: 10.1016/S0140-6736(20)30183-5. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0140673620301835> (visitado 17-05-2020).
- Isea, Raúl y Karl E. Lonngren (2013). *On the Mathematical Interpretation of Epidemics by Kermack and McKendrick*. URL: [http://www.kurims.kyoto-u.ac.jp/EMIS/journals/GMN/yahoo\\_site\\_admin/assets/docs/6\\_GMN-3602-V19N2.32210220.pdf](http://www.kurims.kyoto-u.ac.jp/EMIS/journals/GMN/yahoo_site_admin/assets/docs/6_GMN-3602-V19N2.32210220.pdf).
- Kamikubo, Yasuhiko y Atsushi Takahashi (mar. de 2020). *Epidemiological Tools that Predict Partial Herd Immunity to SARS Coronavirus 2*. en. preprint. Epidemiology. DOI: 10.1101/2020.03.25.20043679. URL: <http://medrxiv.org/lookup/doi/10.1101/2020.03.25.20043679> (visitado 17-05-2020).
- Kermack, William Ogilvy y A. G. McKendrick (ago. de 1927). “A contribution to the mathematical theory of epidemics”. en. En: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 115.772, págs. 700-721. ISSN: 0950-1207, 2053-9150. DOI: 10.1098/rspa.1927.0118. URL: <https://royalsocietypublishing.org/doi/10.1098/rspa.1927.0118> (visitado 17-05-2020).
- Liu, Quan-Hui y col. (dic. de 2018). “Measurability of the epidemic reproduction number in data-driven contact networks”. en. En: *Proceedings of the National Academy of Sciences* 115.50, págs. 12680-12685. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1811115115. URL: <http://www.pnas.org/lookup/doi/10.1073/pnas.1811115115> (visitado 17-05-2020).
- London, Imperial College (2020). *Report 13 - Estimating the number of infections and the impact of non-pharmaceutical interventions on COVID-19 in 11 European countries*. en-GB. URL: <http://www.imperial.ac.uk/medicine/departments/school-public-health/infectious-disease-epidemiology/mrc-global-infectious-disease-analysis/covid-19/report-13-europe-npi-impact/> (visitado 17-05-2020).
- Martín-Corral y col. (2020). *COVID-19 Social Distancing Strategies*. en. URL: <https://covid-19-sds.github.io/> (visitado 17-05-2020).
- McCluskey, C. Connell (feb. de 2010). “Complete global stability for an SIR epidemic model with delay — Distributed or discrete”. en. En: *Nonlinear Analysis: Real World Ap-*

*plications* 11.1, págs. 55-59. ISSN: 1468-1218. DOI: 10.1016/j.nonrwa.2008.10.014. URL: <http://www.sciencedirect.com/science/article/pii/S1468121808002447> (visitado 17-05-2020).

OMS (2020). *Blueprint for R&D preparedness and response to public health emergencies due to highly infectious pathogens*. en. URL: <https://www.who.int/who-documents-detail/blueprint-for-r-d-preparedness-and-response-to-public-health-emergencies-due-to-highly-infectious-pathogens> (visitado 17-05-2020).

Rascado Sedes, Pedro y MaríaÁngeles Ballesteros Sanz (2020). “PLAN DE CONTINGENCIA PARA LOS SERVICIOS DE MEDICINA INTENSIVA FRENTE A LA PANDEMIA COVID-19”. En: Sociedad Española de Medicina Intensiva, Crítica y Unidades Coronarias (SEMICYUC).

Weiss, Howard (Howie) (2020). “The SIR modeland theFoundations of Public Health”. En: *MATerials MATemàtics, Publicació electrònica de divulgació del Departament de Matemàtiquesde la Universitat Autònoma de Barcelona* 2013.treball 3, pág. 17. ISSN: 1887-1097. URL: [https://ddd.uab.cat/pub/matmat/matmat\\_a2013/matmat\\_a2013a3.pdf](https://ddd.uab.cat/pub/matmat/matmat_a2013/matmat_a2013a3.pdf).

WHO (2020). *WHO / WHO publishes list of top emerging diseases likely to cause major epidemics*. URL: <http://www.who.int/medicines/ebola-treatment/WHO-list-of-top-emerging-diseases/en/> (visitado 17-05-2020).

## A. Licencia

Este paper está disponible bajo los principios del Open Access / Acceso Libre, y sus usuarios tienen derecho a: **compartir**: copiar y redistribuir el material en cualquier medio o formato; y a **adaptar**: modificar, transformar, y derivar el material para cualquier propósito



Siempre y cuando se atribuya a los autores de forma adecuada, y se añada un enlace a la licencia, que es CC-BY-Sa 4.0, indicando los cambios realizados, y sin sugerir que los autores del presente estudio apoyan o han colaborado en la realización del mismo.

Queda prohibido aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros de hacer cualquier cosa que la licencia permita.

## B. Visión global de la epidemia

B.1. Lámina 1: Confinamiento efectivo el 17/03/2020

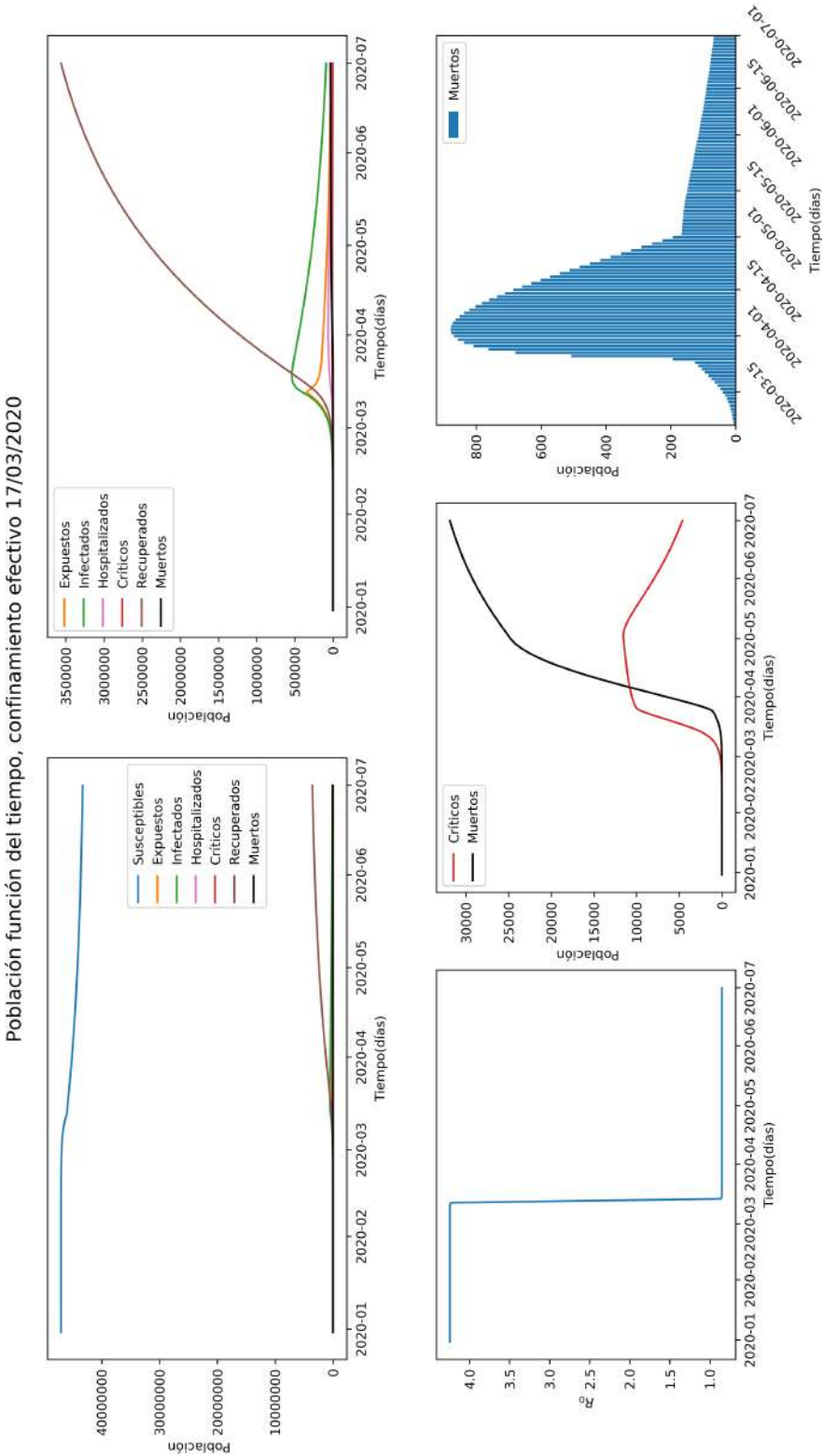


Figura 9: En orden: comportamiento de los grupos de población (1, 2), variación de  $R_0$  (3), evolución de muertes y de personas en UCI (4), tasa diaria de fallecidos(5)



B.2. Lámina 2: confinamiento efectivo el 10/03/2020

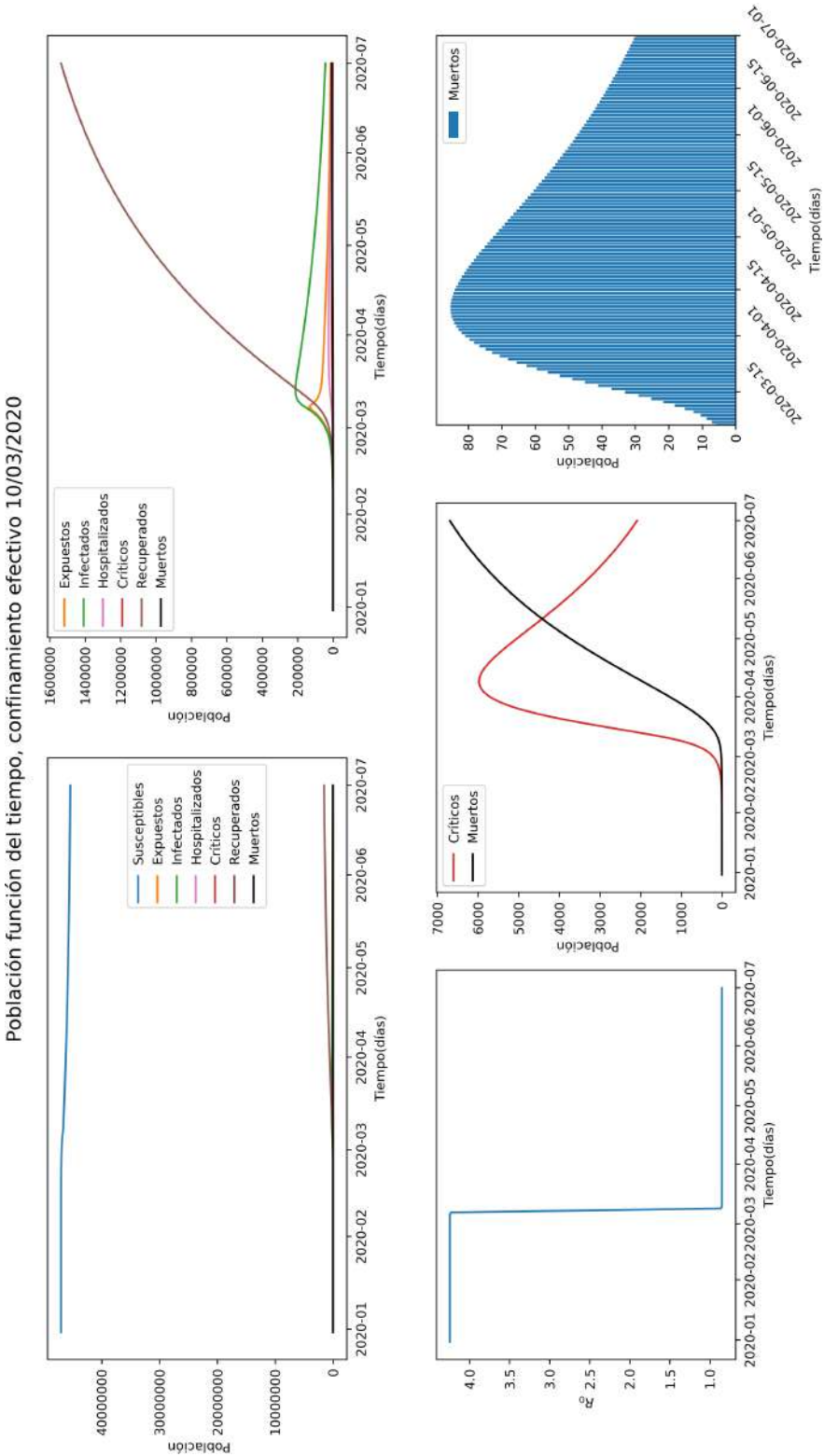


Figura 10: En orden: comportamiento de los grupos de población (1, 2), variación de  $R_0$  (3), evolución de muertes y de personas en UCI (4), tasa diaria de fallecidos(5)

### B.3. Lámina 3: confinamiento efectivo el 24/03/2020

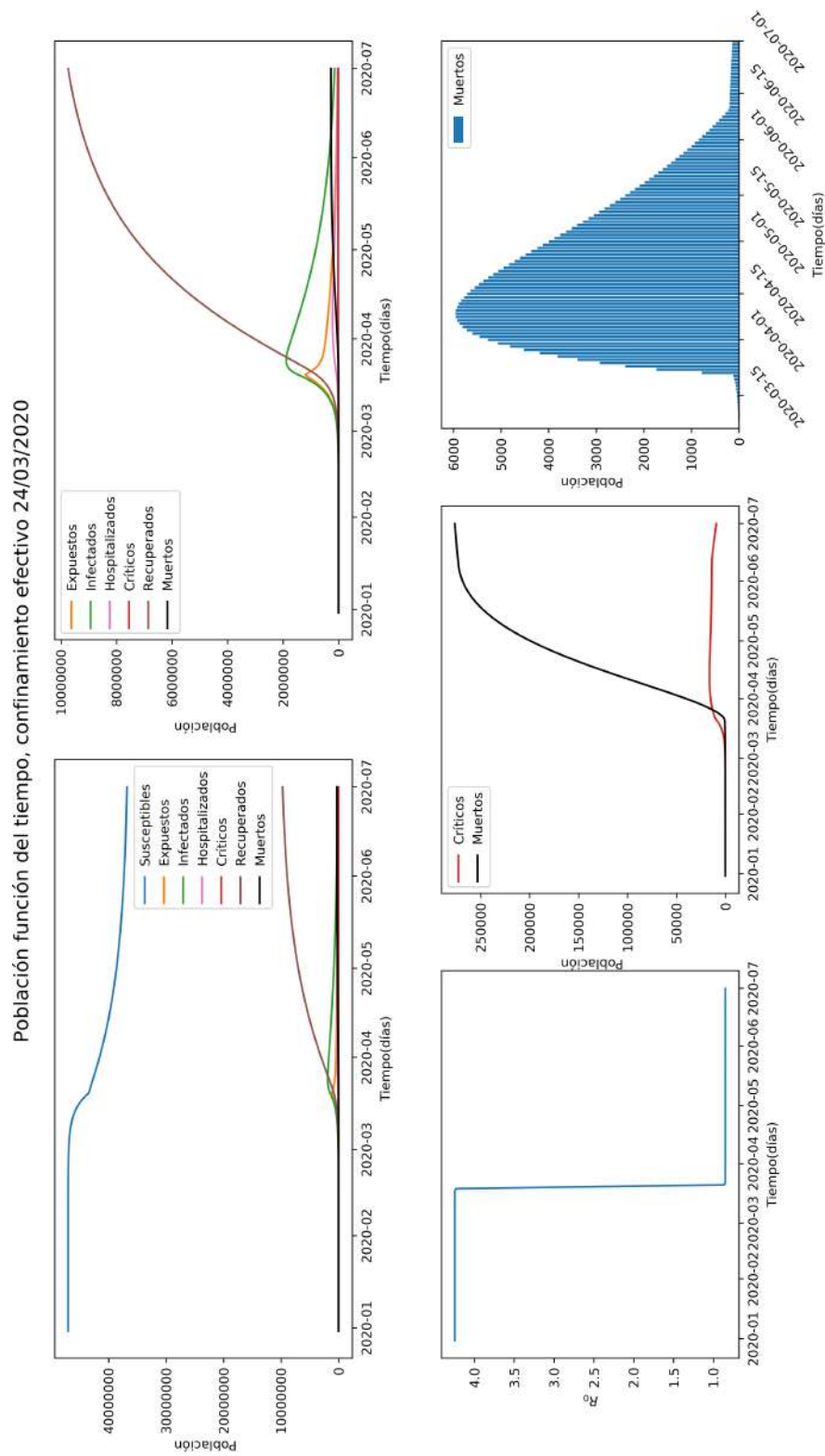


Figura 11: En orden: comportamiento de los grupos de población (1, 2), variación de  $R_0$  (3), evolución de muertes y de personas en UCI (4), tasa diaria de fallecidos(5)

B.4. Lámina 4: Sin confinamiento

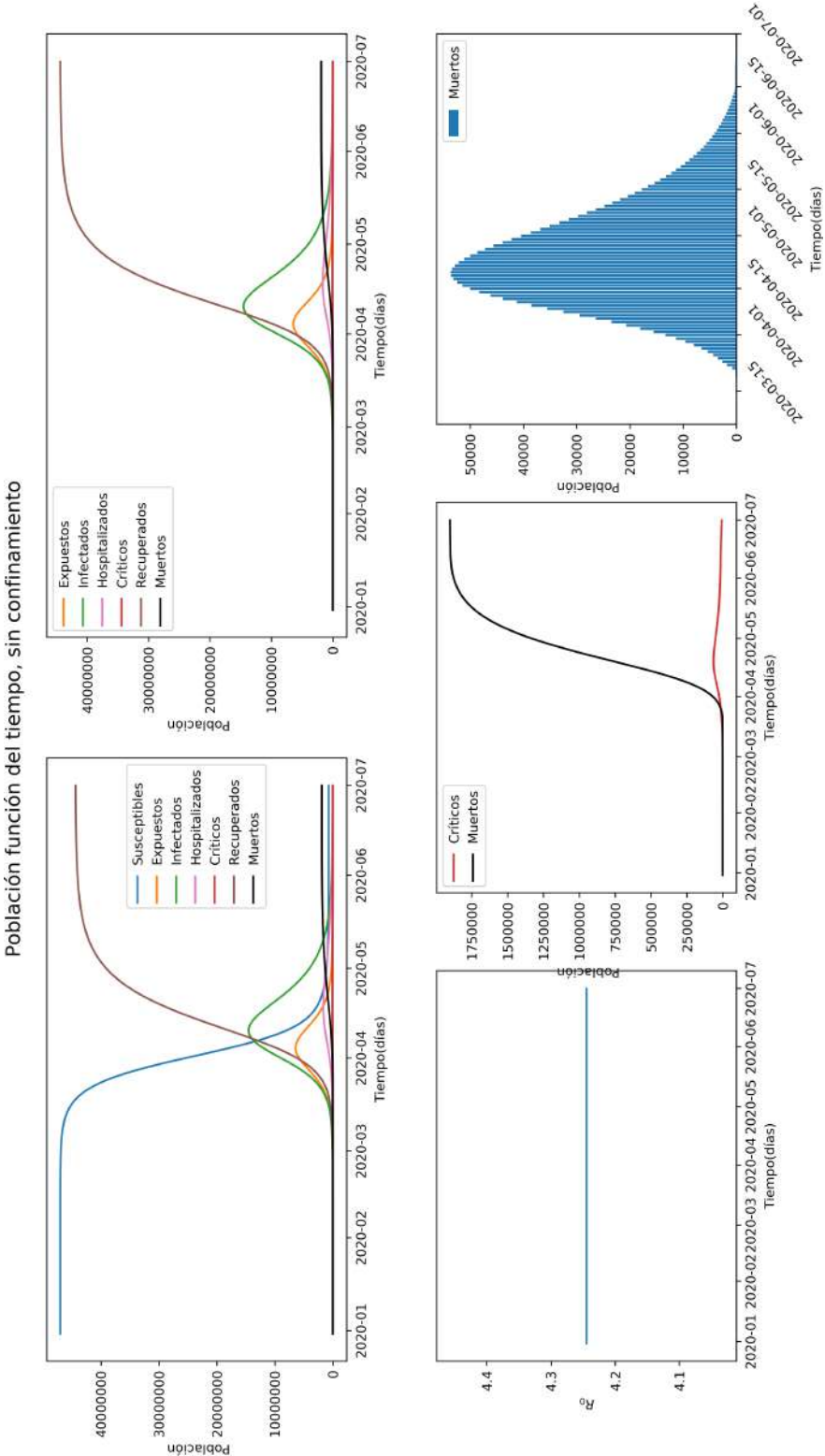


Figura 12: En orden: comportamiento de los grupos de población (1, 2), variación de  $R_0$  (3), evolución de muertes y de personas en UCI (4), tasa diaria de fallecidos(5)

## C. Métodos Matemáticos

### C.1. Modelo SIRS

#### SIRS

May 21, 2020

#### 1 Modelo de SIRS para el estudio de enfermedades

Se establecen tres categorías: susceptibles, infectados y ‘recovered’ (se han recuperado pero pueden volver a ser susceptibles, a diferencia del SIR). Las ecuaciones son:

$$\begin{aligned}\frac{dS}{dt} &= -\beta IS + \nu R \\ \frac{dI}{dt} &= \beta IS - \gamma I \\ \frac{dR}{dt} &= \gamma I - \nu R\end{aligned}$$

Siguiendo indicaciones de página 247 (292 en el PDF) se reformulan estas 3 ecuaciones para dejarlas en 2:

$$\begin{aligned}\frac{dS}{dt} &= -IS\beta + \nu(-I + N - S) \\ \frac{dI}{dt} &= IS\beta - I\gamma\end{aligned}$$

Tasas:

$\beta$  : número de personas infectadas por individuo enfermo y día.

$\gamma$  : 1/período de infección. Es la tasa a la que se recuperan los infectados.

$\nu$  : 1/período de pérdida de inmunidad. Es la tasa a la que los individuos recuperados pierden la inmunidad. En este modelo se asume que la inmunidad es temporal y, por tanto, los individuos recuperados pueden volver a contraer la enfermedad.

Grupos de población:

S : susceptibles de contraer la enfermedad

I : infectados, contagian la enfermedad

R : han superado la enfermedad, no contagian el virus y han desarrollado inmunidad

```
[2]: import sympy as sp
import numpy as np
import scipy.integrate as scint
import matplotlib.pyplot as plt
```

```
sp.init_printing()
```

```
[3]: N, S, I, R, b, v, g = sp.symbols('N S I R      ')
```

```
[4]: f = -b*I*S + v*(N-S-I)
     g = b*I*S-g*I
     f, g
```

```
[4]:  $(-IS + (-I + N - S), IS - I)$ 
```

```
[5]: ptos_criticos = sp.solve([f, g], [S, I])
     p1 = ptos_criticos[0]
     p2_pre = ptos_criticos[1]
     p1, p2_pre
```

```
[5]:  $\left( (N, 0), \left( -\frac{-N + (N-)}{(+)} , \frac{(N-)}{(+)} \right) \right)$ 
```

Por razones desconocidas, la componente X del punto 2 no está correctamente simplificada, así que se procede a una simplificación:

```
[6]: p2 = sp.simplify(p2_pre)
     p2
```

```
[6]:  $\left( , \frac{(N-)}{(+)} \right)$ 
```

De todas formas, a efectos del cálculo del resto de elementos es irrelevante, pero nos va a servir más adelante para definir el número básico de reproducción. El punto (N,0) establece cuando la población está sana y el virus ausente. El otro punto describe el equilibrio al que tiende la población de individuos susceptibles y sanos.

```
[7]: J = sp.Matrix([[f.diff(S), f.diff(I)], [g.diff(S), g.diff(I)]])
     J
```

```
[7]:  $\begin{bmatrix} -I - & -S - \\ I & S - \end{bmatrix}$ 
```

Punto 0,0

```
[8]: J1 = J.subs({S:p1[0], I:p1[1]})
     J1
```

```
[8]:  $\begin{bmatrix} - & -N - \\ 0 & N - \end{bmatrix}$ 
```

```
[9]: autovectores1=J1.eigenvects()
     autovectores1
```

```
[9]:
```

$$\left[ \left( -, 1, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right), \left( N-, 1, \begin{bmatrix} -\frac{N-}{-N+-} \\ 1 \end{bmatrix} \right) \right]$$

Punto (0, S)

```
[10]: J2 = sp.simplify(J.subs({S:p2[0], I:p2[1]}))
      J2
```

```
[10]: 
$$\begin{bmatrix} -\frac{(N+)}{(N+)} & - \\ \frac{(N-)}{+} & 0 \end{bmatrix}$$

```

```
[11]: autovectores2=J2.eigenvecs()
      autovectores2
```

```
[11]: 
$$\left[ \left( -\frac{(N+)}{2(+)} - \frac{\sqrt{(N^{22}-4N^2-8N-2N^2+4^3+8^2+4^2+3)}}{2(+)}, 1, \begin{bmatrix} -\frac{-}{-} \\ -\frac{(N+)}{2(+)} + \frac{\sqrt{(N^{22}-4N^2-8N-2N^2+4^3+8^2+4^2+3)}}{2(+)} \end{bmatrix} \right) \right]$$

```

```
[12]: sp.simplify(J2.det())
```

```
[12]: (N-)
```

```
[13]: sp.simplify(J2.trace())
```

```
[13]: 
$$-\frac{(N+)}{+}$$

```

Existen condiciones de estabilidad para los cuales los autovalores son complejos.

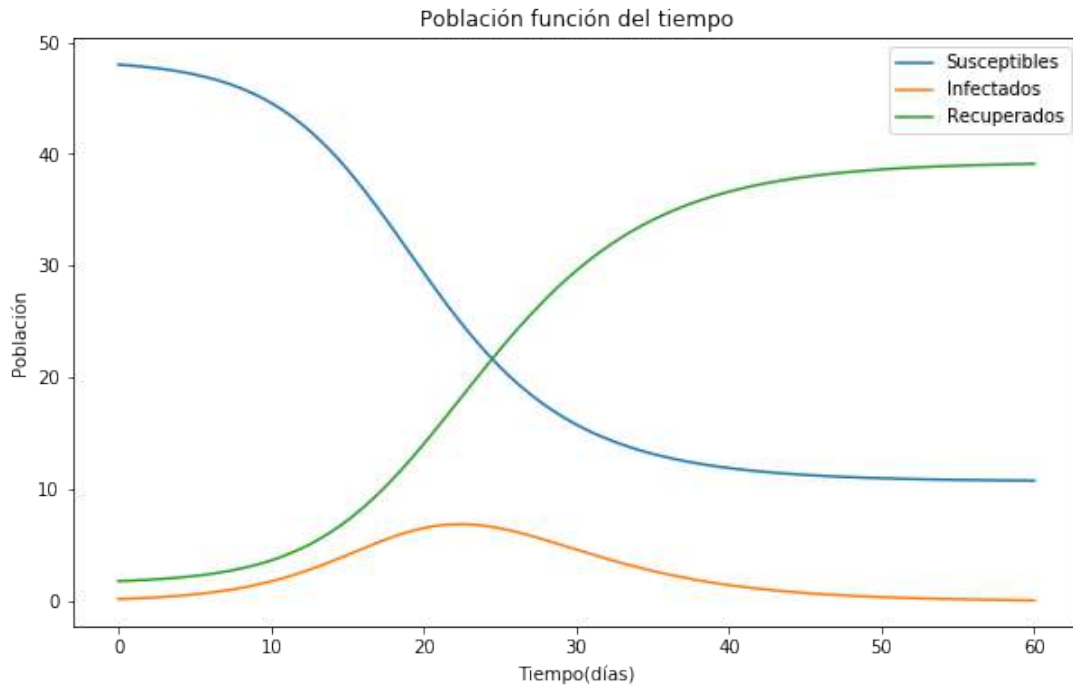
Visualización

```
[14]: def F(P, t):
      return [-b*P[1]*P[0] + g*(N-P[0]-P[1]), b*P[1]*P[0]-v*P[1]]
```

```
[23]: b, v, g, N = 0.01,0.25,0,50
      tiempo=np.linspace(0, 60, 1000)
      P0 = [48, 0.2]
      Ps = scint.odeint(F, P0, tiempo)

      susceptibles = Ps[:,0]
      infectados = Ps[:,1]
      poblacionini = 50*np.ones(1000)
      recuperados = poblacionini-susceptibles-infectados
      plt.figure(figsize = (10,6))
      plt.plot(tiempo, susceptibles, "-", label = "Susceptibles")
      plt.plot(tiempo, infectados, "-", label = "Infectados")
      plt.plot(tiempo, recuperados, "-", label = "Recuperados")
      plt.xlabel("Tiempo(días)")
      plt.ylabel("Población")
      plt.title("Población función del tiempo")
      plt.legend()
```

```
plt.savefig('SIRSpoblaciontiempo.png', dpi = 300, bbox_inches = 'tight', pad_inches = 0)
plt.show()
```



```
[16]: def analisis_global(b, v, g, N,
        dimp = 0.2,
        dims = 5,
        pm = 20,
        autovector=(autovectores1, autovectores2)):

    def F(P, t):
        return [-b*P[1]*P[0] + g*(N-P[0]-P[1]), b*P[1]*P[0]-v*P[1]]

    def campo_vectorial(xmin, xmax, ymin, ymax, pm, tiempo):
        X, Y = np.meshgrid(np.linspace(xmin, xmax, pm), np.linspace(ymin,
        ymax, pm))
        U = F([X, Y], tiempo)[0]
        V = F([X, Y], tiempo)[1]
        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 = 30)

def autovec(autovector, dim, xo, yo, b_=b, v_=v, g_=g, N_=N):
    N, S, I, R, b, v, g = sp.symbols('N S I R      ')
    autovector1=autovector[0][2][0][0], autovector[0][2][0][1]
    autovector2=autovector[1][2][0][0], autovector[1][2][0][1]
    autovectorx=autovector1[1]/autovector1[0]
    autovectory=autovector2[1]/autovector2[0]

    dirx = sp.lambdify([b, v, g, N],autovectorx)
    diry = sp.lambdify([b, v, g, N],autovectory)

    b, v, g, N = b_, v_, g_, N_
    x = np.linspace(-dim + xo, dim + xo, 1000)
    y1= dirx(b, v, g, N)*(x-xo) + yo
    y2= diry(b, v, g, N)*(x-xo) + yo
    if autovector1[0]==0:
        plt.plot((xo,xo),(-dim+yo,dim+yo),color = 'r', linewidth=3)
    if autovector2[0]==0:
        plt.plot((xo,xo),(-dim,dim),color = 'r', linewidth=3)
    plt.plot(x, y1, color = 'r', linewidth=3)
    plt.plot(x, y2,color = 'r', linewidth=3)

def pcriticos(x,y, tiempo, dimx = 0.1, dimy = 0.1, pm =20):
    dim=dimx
    campo_vectorial(-dim+x,dim+x,-dim+y,dim+y,pm, tiempo)
    ic = np.linspace(-dim, dim, pm)
    for r in ic:
        X0 = [r +x, dim +y]
        Xs = scint.odeint(F, X0, tiempo)
        plt.plot(Xs[:,0], Xs[:,1], "-")
    for s in ic:
        X0 = [-dim+x, s+y]
        Xs = scint.odeint(F, X0, tiempo)
        plt.plot(Xs[:,0], Xs[:,1], "-")
    for r in ic:
        X0 = [r+x, -dim+y]
        Xs = scint.odeint(F, X0, tiempo)
        plt.plot(Xs[:,0], Xs[:,1], "-")
    for s in ic:
        X0 = [dim+x, s+y]
        Xs = scint.odeint(F, X0, tiempo)

```



```

plt.plot(Xs[:,0], Xs[:,1], "-")
plt.xlim(-dim+x, dim+x)
plt.ylim(-dim+y, dim+y)

plt.figure(figsize=(16,8))
p1 = (N,0)
p2 = (v/g, g*(N*b-v)/(b*(g+v)))
tiempo1 = np.flip(np.linspace(0, pm, 1000))
tiempo2 = np.linspace(0, pm, 1000)

plt.subplot(1,2,1)
pcriticos(p1[0],p1[1],tiempo1, dimp, pm)
autovec(autovector[0], dimp, p1[0], p1[1], b, v, g, N)
plt.scatter(p1[0],p1[1], color = 'blue', marker = 'D',zorder=10)
plt.title('Punto $(N,0)$')

plt.subplot(1,2,2)
pcriticos(p2[0],p2[1], tiempo2, dimp, pm)
autovec(autovector[1], dimp, p2[0], p2[1], b, v, g, N)
plt.scatter(p2[0],p2[1], color = 'blue', marker = 'D',zorder=10)
plt.title('Punto $(S^*_2, I^*_2)$')

```

```

[17]: analisis_global(1,2,1,5)
plt.savefig('puntosdeequilibrio.png', dpi = 300, bbox_inches =
↳ 'tight',pad_inches = 0)

```

```

/home/pablo/anaconda3/lib/python3.7/site-
packages/scipy/integrate/odepack.py:247: ODEintWarning: Excess work done on this
call (perhaps wrong Dfun type). Run with full_output = 1 to get quantitative
information.

```

```

warnings.warn(warning_msg, ODEintWarning)

```

```

/home/pablo/anaconda3/lib/python3.7/site-packages/matplotlib/axes/_base.py:2485:
RuntimeWarning: overflow encountered in double_scalars

```

```

x1t += delta

```

```

/home/pablo/anaconda3/lib/python3.7/site-packages/numpy/core/_asarray.py:85:

```

```

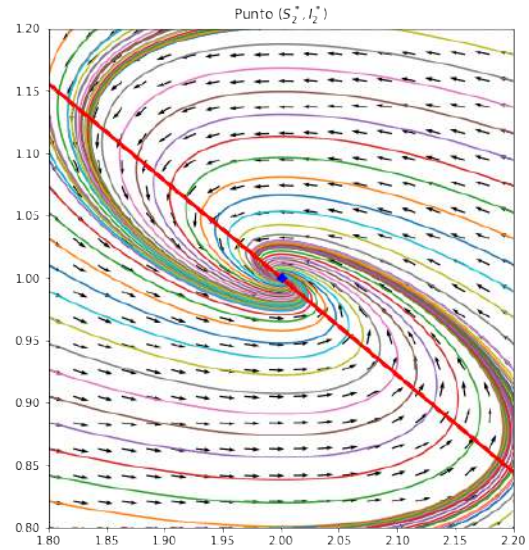
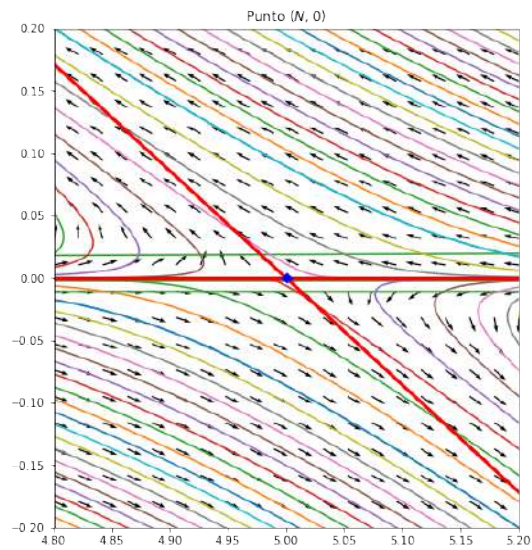
ComplexWarning: Casting complex values to real discards the imaginary part

```

```

return array(a, dtype, copy=False, order=order)

```



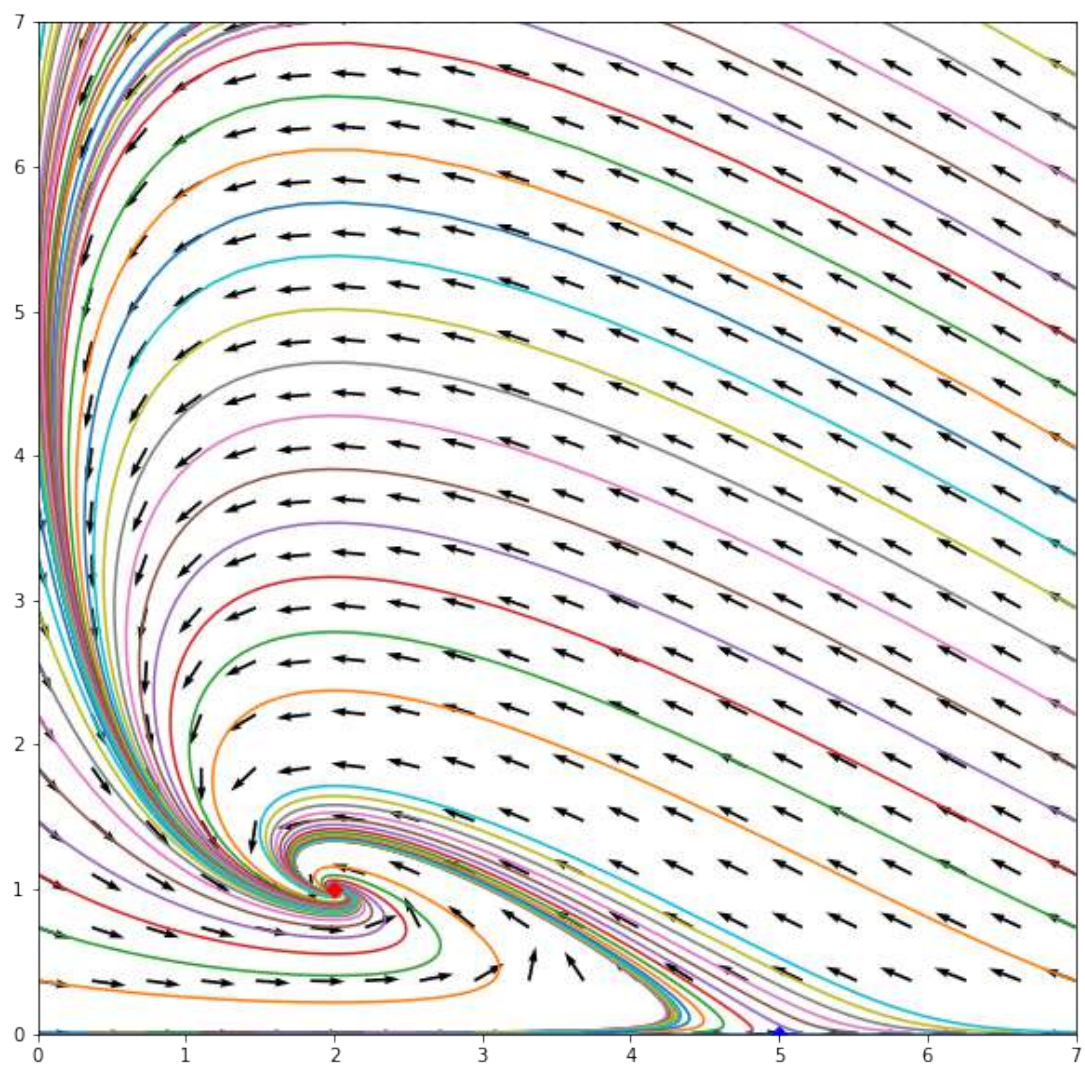
```
[18]: b, v, g, N = 1,2,1,5
plt.figure(figsize=(10,10))
xmin, xmax, ymin, ymax = 0, 7, 0, 7
dim = 7
pm=20
tiempo=np.linspace(0,20,1000)
X, Y = np.meshgrid(np.linspace(xmin, xmax, pm), np.linspace(ymin, ymax, pm))
U = F([X, Y], tiempo)[0]
V = F([X, Y], tiempo)[1]
modulo2 = U**2 + V**2
D1 = np.ones([pm, 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 = 30)

ic = np.linspace(0,7, pm)
for r in ic:
    X0 = [r, 0.01]
    Xs = scint.odeint(F, X0, tiempo)
    plt.plot(Xs[:,0], Xs[:,1], "-")
for s in ic:
```

```

X0 = [0, s]
Xs = scint.odeint(F, X0, tiempo)
plt.plot(Xs[:,0], Xs[:,1], "-")
for r in ic:
    X0 = [r, 7]
    Xs = scint.odeint(F, X0, tiempo)
    plt.plot(Xs[:,0], Xs[:,1], "-")
for s in ic:
    X0 = [7, s]
    Xs = scint.odeint(F, X0, tiempo)
    plt.plot(Xs[:,0], Xs[:,1], "-")
plt.xlim(0,7)
plt.ylim(0,7)
p1 = (N,0)
p2 = (v/g, g*(N*b-v)/(b*(g+v)))
plt.scatter(p1[0],p1[1], color = 'blue', marker = 'D',zorder=10)
plt.scatter(p2[0],p2[1], color = 'red', marker = 'D',zorder=10)
plt.savefig('sistemaglobalSIRS.png', dpi = 300, bbox_inches =
↳ 'tight',pad_inches = 0)

```



## C.2. Modelo SEIR Modificado

### SEIR\_modified

May 21, 2020

## 1 Modelo de estudio de la pandemia causada por SARS-Coronavirus-2

Estos individuos necesitan cuidados intensivos, por lo que es una forma de medir la intervención del sistema nacional de salud en el control de la pandemia. Asumimos que todos aquellos individuos críticos que no reciben asistencia sanitaria mueren.

$$\begin{aligned}\frac{dS}{dt} &= -\beta I \frac{S}{N} \\ \frac{dE}{dt} &= \beta I \frac{S}{N} - \delta E \\ \frac{dI}{dt} &= \delta E - (1-a) \gamma I - \eta a I \\ \frac{dH}{dt} &= \eta (1-a) I - \tau (1-u) H - u \sigma H \\ \frac{dC}{dt} &= u \sigma H - m \rho \min(UCI, C) - \max(0, C - UCI) - \omega (1-m) \min(UCI, C) \\ \frac{dR}{dt} &= \gamma (1-a) I + \tau (1-u) H + \omega (1-m) \min(UCI, C) \\ \frac{dM}{dt} &= m \rho \min(UCI, C) + \max(0, C - UCI)\end{aligned}$$

Tasas:

$\beta$  : número de personas infectadas por individuo enfermo y día.

$\delta$  : 1/periodo de incubación. La tasa a la que los individuos que incuban el virus desarrollan síntomas y pasan a ser infecciosos.

$\gamma$  : 1/periodo de recuperación de casos leves -

$\eta$  : 1/periodo de desarrollo de síntomas que requieren ingreso hospitalario.

$\tau$  : 1/periodo de estancia hospitalaria hasta recuperación en casos que no requieren ingreso en UCI.

$\sigma$  : 1/periodo de desarrollo de síntomas que requieren ingreso en UCI.

$\rho$  : 1/tiempo de estancia en UCI hasta la muerte. (7.5)

$\omega$  : 1/tiempo de estancia en UCI hasta recuperación. (6.5)

Probabilidades:

a : probabilidad de que un individuo infectado desarrolle síntomas que requieran ingreso hospitalario.

u : probabilidad de que un individuo infectado pase a estar crítico.

m : probabilidad de que un individuo en estado crítico fallezca.

Grupos de población:

S : susceptibles de contraer la enfermedad

E : expuestos a individuos infectados, incuban la enfermedad pero aún no son contagiosos.

I : infectados, contagian la enfermedad

R : han superado la enfermedad, no contagian el virus y han desarrollado inmunidad.

C : individuos infectados que requieren cuidados intensivos.

M : fallecidos por COVID-19

```
[1]: import sympy as sp
import numpy as np
import scipy.integrate as scint
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import pandas as pd
!pip install lmfit
import lmfit
sp.init_printing()
```

```
Requirement already satisfied: lmfit in
/home/pablo/anaconda3/lib/python3.7/site-packages (1.0.1)
Requirement already satisfied: numpy>=1.16 in
/home/pablo/anaconda3/lib/python3.7/site-packages (from lmfit) (1.17.2)
Requirement already satisfied: asteval>=0.9.16 in
/home/pablo/anaconda3/lib/python3.7/site-packages (from lmfit) (0.9.18)
Requirement already satisfied: scipy>=1.2 in
/home/pablo/anaconda3/lib/python3.7/site-packages (from lmfit) (1.3.1)
Requirement already satisfied: uncertainties>=3.0.1 in
/home/pablo/anaconda3/lib/python3.7/site-packages (from lmfit) (3.1.2)
```

```
[2]: def F(P, t, N, b, m, u, a, UCI):
    S, E, I, H, C, R, M = P
    delta = 1/3
    gamma = 1/9.0
    sigma = 1/4.5
    rho = 1/14
    omega = 1/6.5
    eta = 1/9.0
    tau = 1/14
```

```

dS = -b(t) * S * I / N
dE = b(t) * S * I / N - delta * E
dI = delta * E - I*eta*a - I*gamma*(1-a)
dH = eta*a*I-tau*(1-u)*H-u*sigma*H
dC = u*sigma*H - m*rho*min(UCI(t),C) -
↪max(0,C-UCI(t))-omega*(1-m)*min(UCI(t),C)
dR = I*gamma*(1-a) + tau*(1-u)*H + omega*(1-m)*min(UCI(t),C)
dM = m*rho*min(UCI(t),C) + max(0,C-UCI(t))
return dS, dE, dI, dH, dC, dR, dM

```

```

[3]: #Datos España
datosCOVID19= pd.read_excel(r'Data/COVID-19-geographic-disbtribution-worldwide.
↪xlsx', parse_dates=["dateRep"], skiprows=[1])
demografia = pd.read_excel(r'Data/GruposEdadINE.xlsx')

```

```

[4]: datosCOVID19

```

```

[4]:
   dateRep  day  month  year  cases  deaths  cumulatedeaths  \
0  2020-05-16   16     5   2020   1063     32             NaN
1  2020-05-15   15     5   2020    113      6             NaN
2  2020-05-14   14     5   2020    259      3             NaN
3  2020-05-13   13     5   2020    280      5             NaN
4  2020-05-12   12     5   2020    285      2             NaN
...      ...  ...    ...    ...    ...    ...
17571 2020-03-25   25     3   2020      0      0             NaN
17572 2020-03-24   24     3   2020      0      1             NaN
17573 2020-03-23   23     3   2020      0      0             NaN
17574 2020-03-22   22     3   2020      1      0             NaN
17575 2020-03-21   21     3   2020      1      0             NaN

   countriesAndTerritories  geoId  countryterritoryCode  popData2018  \
0                Afghanistan    AF                AFG    37172386.0
1                Afghanistan    AF                AFG    37172386.0
2                Afghanistan    AF                AFG    37172386.0
3                Afghanistan    AF                AFG    37172386.0
4                Afghanistan    AF                AFG    37172386.0
...      ...      ...
17571                Zimbabwe    ZW                ZWE    14439018.0
17572                Zimbabwe    ZW                ZWE    14439018.0
17573                Zimbabwe    ZW                ZWE    14439018.0
17574                Zimbabwe    ZW                ZWE    14439018.0
17575                Zimbabwe    ZW                ZWE    14439018.0

   continentExp
0             Asia
1             Asia
2             Asia

```



● ● ●                      ● ●

```
[17576 rows x 12 columns]
```

demografia

0	0-9	4394157	0.003
1	10-19	4813817	0.004
2	20-29	4871664	0.002
3	30-39	6352990	0.003
4	40-49	7832487	0.006
5	50-59	6904789	0.015
6	60-69	5197906	0.048
7	70-79	3793879	0.140
8	80-89	2329061	0.210
9	90 +	535458	0.220

```
grupos = demografia["Pob"].values
letalidad = demografia["Letalidad"].values
sum(grupos), letalidad
```

```
(47026208,
 array([0.003, 0.004, 0.002, 0.003, 0.006, 0.015, 0.048, 0.14 , 0.21 ,
        0.22 ]))
```

```
#Condiciones iniciales
UCI_100 = 9.7
def R_0(t, R_Oi, R_Of, k, A):
    return (R_Oi-R_Of) / (1 + np.exp(-k*(-t+A))) + R_Of
```

```
muertesES=datosCOVID19[datosCOVID19["countriesAndTerritories"] ==  
  ↪ "Spain"] ["cumulatedeaths"].values[:: -1]  
  
muertesES
```

```
array([0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
       0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
       0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
       0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
```



```

0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 1.0000e+00, 3.0000e+00, 5.0000e+00, 5.0000e+00,
5.0000e+00, 2.8000e+01, 3.5000e+01, 4.7000e+01, 8.4000e+01,
1.2100e+02, 1.3600e+02, 2.8800e+02, 3.0900e+02, 4.9100e+02,
5.9800e+02, 7.6700e+02, 1.0020e+03, 1.3260e+03, 1.7200e+03,
2.1820e+03, 2.6960e+03, 3.4340e+03, 4.0890e+03, 4.8580e+03,
5.6900e+03, 6.5280e+03, 7.3400e+03, 8.1890e+03, 9.0530e+03,
1.0003e+04, 1.0935e+04, 1.1744e+04, 1.2418e+04, 1.3055e+04,
1.3798e+04, 1.4555e+04, 1.5238e+04, 1.5843e+04, 1.6353e+04,
1.6972e+04, 1.7489e+04, 1.8276e+04, 1.8893e+04, 1.9478e+04,
2.0043e+04, 2.0453e+04, 2.0852e+04, 2.1282e+04, 2.1717e+04,
2.2157e+04, 2.2524e+04, 2.2902e+04, 2.3190e+04, 2.3190e+04,
2.3822e+04, 2.4275e+04, 2.4543e+04, 2.4824e+04, 2.5100e+04,
2.5264e+04, 2.5428e+04, 2.5613e+04, 2.5857e+04, 2.6070e+04,
2.6251e+04, 2.6478e+04, 2.6621e+04, 2.6744e+04, 2.6920e+04,
2.7104e+04, 2.7321e+04, 2.7459e+04, 2.7563e+04])

```

```
[9]: datosCOVID19[datosCOVID19["countriesAndTerritories"] == "Spain"][:64]
```

```

[9]:      dateRep  day  month  year  cases  deaths  cumulatedeaths  \
15033 2020-05-17   17     5   2020   421     87           27563.0
15034 2020-05-16   16     5   2020   515    104           27459.0
15035 2020-05-15   15     5   2020   643    138           27321.0
15036 2020-05-14   14     5   2020   849    217           27104.0
15037 2020-05-13   13     5   2020   439    184           26920.0
...      ...  ...  ...  ...  ...  ...  ...
15103 2020-03-08    8     3   2020   330     0              5.0
15104 2020-03-07    7     3   2020   233     2              3.0
15105 2020-03-06    6     3   2020   183     2              1.0
15106 2020-03-05    5     3   2020    81     1              0.0
15107 2020-03-04    4     3   2020    75     0              0.0

      countriesAndTerritories  geoId  countryterritoryCode  popData2018  \
15033                Spain      ES                ESP    46723749.0
15034                Spain      ES                ESP    46723749.0
15035                Spain      ES                ESP    46723749.0
15036                Spain      ES                ESP    46723749.0
15037                Spain      ES                ESP    46723749.0
...      ...  ...  ...  ...  ...

```

15103	Spain	ES	ESP	46723749.0
15104	Spain	ES	ESP	46723749.0
15105	Spain	ES	ESP	46723749.0
15106	Spain	ES	ESP	46723749.0
15107	Spain	ES	ESP	46723749.0

	continentExp
15033	Europe
15034	Europe
15035	Europe
15036	Europe
15037	Europe
...	...
15103	Europe
15104	Europe
15105	Europe
15106	Europe
15107	Europe

[75 rows x 12 columns]

```
[10]: def modelo(dias, grupos, UCI_100, R_Oi, R_Of, k, A, u, m, a, s):
    gamma = 1/9.0
    def b(t):
        return R_0(t, R_Oi, R_Of, k, A) * gamma

    N = sum(grupos)

    def UCI(t):
        UCI_0 = UCI_100 / 100_000 * N
        return UCI_0 + s*UCI_0*t

    P0 = N-1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0
    tiempo = np.linspace(0, dias-1, dias, dtype=int)
    Ps = scint.odeint(F, P0, tiempo, args=(N, b, m, u, a, UCI))
    S = Ps[:,0]
    E = Ps[:,1]
    I = Ps[:,2]
    H = Ps[:,3]
    C = Ps[:,4]
    R = Ps[:,5]
    M = Ps[:,6]
    R_0t = R_0(tiempo, R_Oi, R_Of, k, A)

    return S, E, I, H, C, R, M, R_0t
```

```
[11]: params_init_min_max = {"R_Oi": (3.8, 2.0, 5), "k": (0.234, 0.01, 10),
                             "A": (75,73,77), "R_Of": (1, 0.8, 1.5),
                             "u": (0.15, 0.15, 0.2), "m": (0.05, 0.01, 0.2),
                             "s": (0.009, 0.001, 0.02), "a": (0.11, 0.05, 0.2)}
```

```
[12]: def ajuste(x, A, R_Oi, R_Of, k, u, m, a, s):
        Ps = modelo(dias, grupos, UCI_100, R_Oi, R_Of, k, A, u, m, a, s)
        return Ps[6][x]
```

```
[13]: dias = len(muertesES)
        dias
```

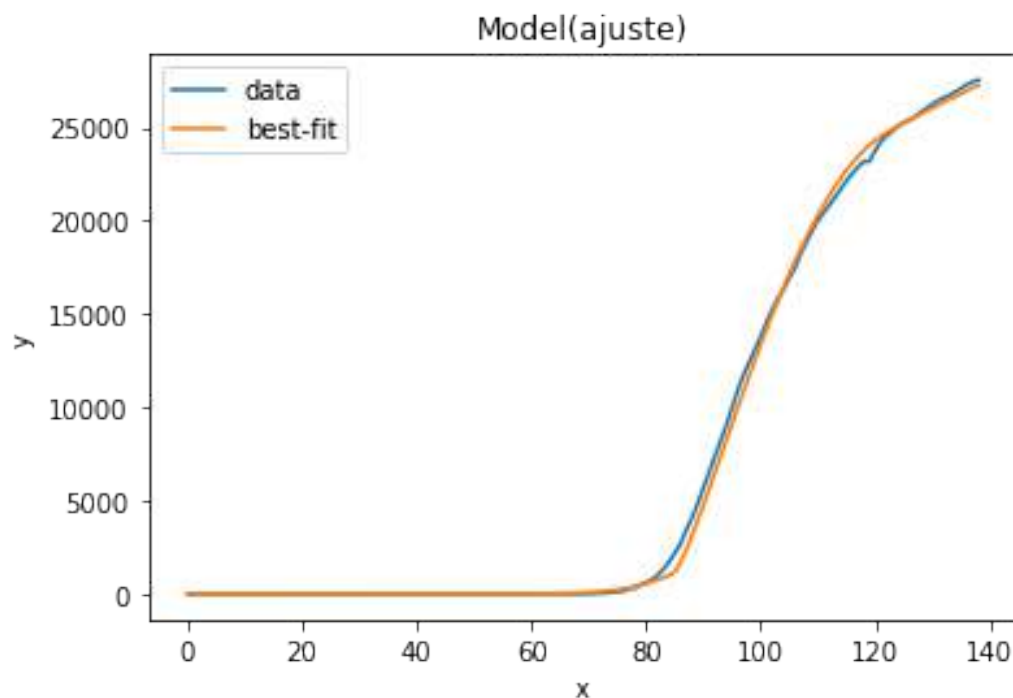
```
[13]: 139
```

```
[14]: compu = lmfit.Model(ajuste)
        for kwarg, (init, mini, maxi) in params_init_min_max.items():
            compu.set_param_hint(str(kwarg), value=init, min=mini, max=maxi, vary=True)

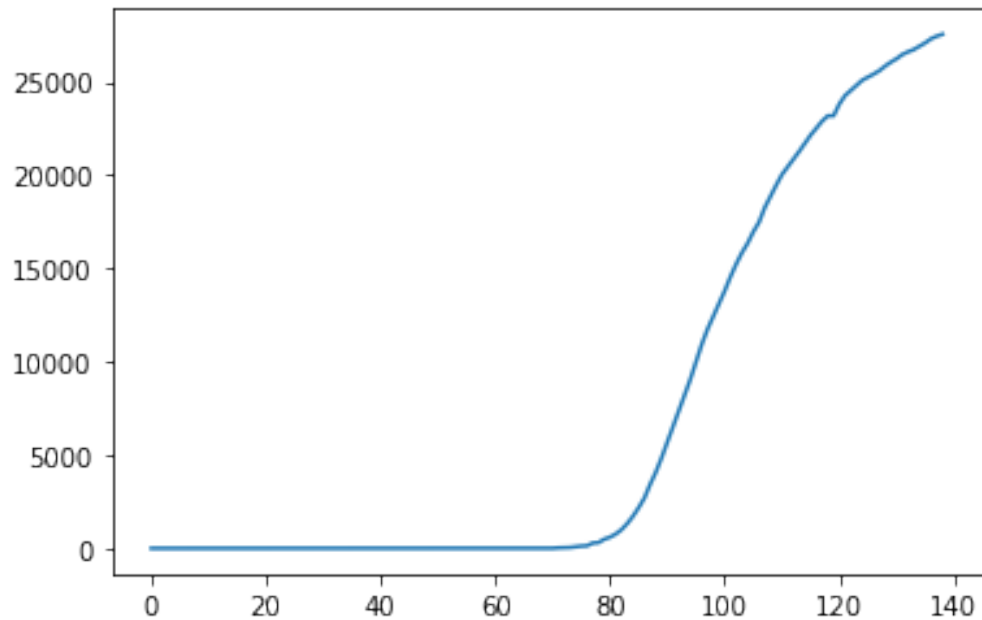
        params = compu.make_params()
        fit_method = "leastsq"
```

```
[15]: periodo = np.linspace(0, dias-1, dias, dtype=int)
        result = compu.fit(muertesES, params, method=fit_method, x=periodo)
```

```
[16]: result.plot_fit(datafmt="-");
```



```
[17]: plt.plot(periodo, muertesES)
plt.show()
```



```
[18]: datos=result.best_values
datos
```

```
[18]: {'A': 73.00000000000001,
'R_Oi': 4.244683874130367,
'R_Of': 0.8486004808522308,
'k': 5.392645813423202,
'u': 0.15,
'm': 0.19992198566259523,
'a': 0.125667277022754,
's': 0.01251477914438805}
```

```
[19]: A = datos['A']
R_Oi=datos['R_Oi']
R_Of=datos['R_Of']
k = datos['k']
u = datos['u']
m = datos['m']
a = datos['a']
s = datos['s']
```

```
[20]: rango = 184
fechas = pd.date_range(start = pd.to_datetime('31/12/2019', dayfirst = True),
                        periods = rango, freq = 'D')
```

```
[21]: mod = modelo(rango, grupos, UCI_100, R_Oi, R_Of, k, A, u, m, a, s)
tiempo = np.linspace(0,rango,dias)
susceptibles = mod[0]
exposeados = mod[1]
infectados = mod[2]
hospitalizados = mod[3]
criticos = mod[4]
recuperados = mod[5]
muertos= mod[6]
R_Ot = mod[7]
muertes_diarias=[0]
for i in range(0, len(muertos)-1):
    muertes_diarias.append(muertos[i+1]-muertos[i])
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.title("A")
plt.plot(fechas, exposeados, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color = '#e377c2')
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xticks(rotation=40)
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.subplot(1,2,2)
plt.title("B")
plt.bar(fechas, muertes_diarias, label = "Muertos")
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xlim(fechas[65], fechas[rango-1])
plt.xticks(rotation=40)
plt.legend()
plt.savefig('1703.png', dpi = 300, bbox_inches = 'tight',pad_inches = 0)
plt.show()

plt.figure(figsize = (18,9))
plt.suptitle("Población función del tiempo, confinamiento efectivo 17/03/2020",
            fontsize = 15)
plt.subplots_adjust(wspace = 0.2, hspace = 0.3,top = 0.93)
plt.subplot(2,2,1)
```

```

plt.plot(fechas, susceptibles, "-", label = "Susceptibles")
plt.plot(fechas, expuestos, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color = '
↳ '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.subplot(2,3,4)
plt.plot(fechas, R_0t)
plt.xlabel("Tiempo(días)")
plt.ylabel("$R_0$")
plt.subplot(2,3,5)
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.subplot(2,3,6)
plt.bar(fechas, muertes_diarias, label = "Muertos")
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xlim(fechas[65], fechas[rango-1])
plt.xticks(rotation=40)
plt.legend()
plt.subplot(2,2,2)
plt.plot(fechas, expuestos, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color = '
↳ '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.savefig('globalconf1703.png', dpi = 300, bbox_inches = 'tight', pad_inches = '
↳ 0)
plt.show()

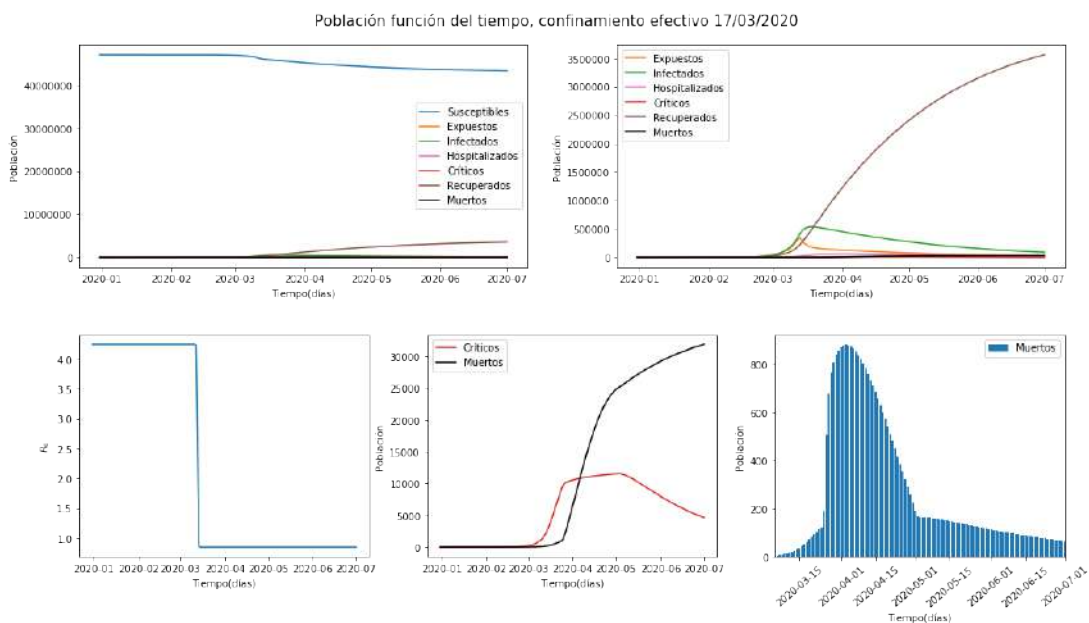
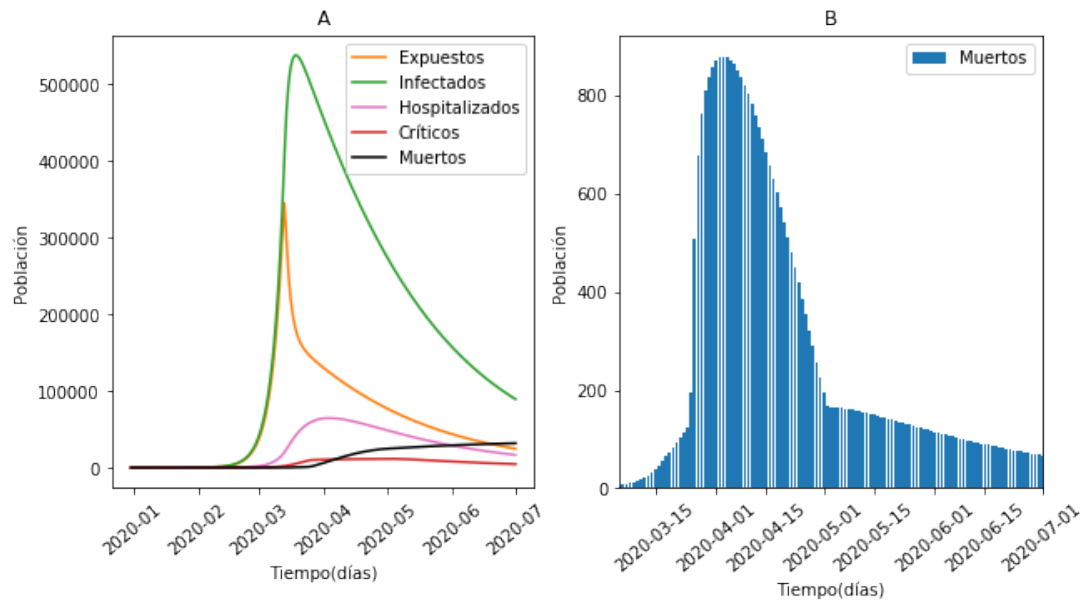
```

/home/pablo/anaconda3/lib/python3.7/site-packages/pandas/plotting/\_matplotlib/converter.py:103: FutureWarning: Using an

implicitly registered datetime converter for a matplotlib plotting method. The converter was registered by pandas on import. Future versions of pandas will require you to explicitly register matplotlib converters.

To register the converters:

```
>>> from pandas.plotting import register_matplotlib_converters
>>> register_matplotlib_converters()
warnings.warn(msg, FutureWarning)
```



```
[22]: panorama=(susceptibles[123],  
↳exposeados[123],infectados[123],hospitalizados[123],  
↳criticos[123],recuperados[123],muertos[123])  
panorama, sum(panorama)
```

```
[22]: ((44161108.68095498, 75603.80748345947, 269282.9207749833, 47876.65971642243, 11558.5792532785, 243585
```

```
[23]: mod = modelo(rango, grupos, UCI_100, R_Oi, R_Of, k, 68, u, m, a, s)  
tiempo = np.linspace(0,rango,dias)  
susceptibles = mod[0]  
exposeados = mod[1]  
infectados = mod[2]  
hospitalizados = mod[3]  
criticos = mod[4]  
recuperados = mod[5]  
muertos= mod[6]  
R_Ot = mod[7]  
muertes_diarias=[0]  
for i in range(0, len(muertos)-1):  
    muertes_diarias.append(muertos[i+1]-muertos[i])  
plt.figure(figsize=(10,5))  
plt.subplot(1,2,1)  
plt.title("A")  
plt.plot(fechas, exposeados, "-", label = "Expuestos", color = '#ff7f0e')  
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')  
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color =  
↳'#e377c2' )  
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')  
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')  
plt.xlabel("Tiempo(días)")  
plt.ylabel("Población")  
plt.xticks(rotation=40)  
plt.legend()  
plt.ticklabel_format(style='plain', axis = 'y')  
plt.subplot(1,2,2)  
plt.title("B")  
plt.bar(fechas, muertes_diarias, label = "Muertos")  
plt.xlabel("Tiempo(días)")  
plt.ylabel("Población")  
plt.xlim(fechas[65], fechas[rango-1])  
plt.xticks(rotation=40)  
plt.legend()  
plt.savefig('1003.png', dpi = 300, bbox_inches = 'tight',pad_inches = 0)  
plt.show()
```

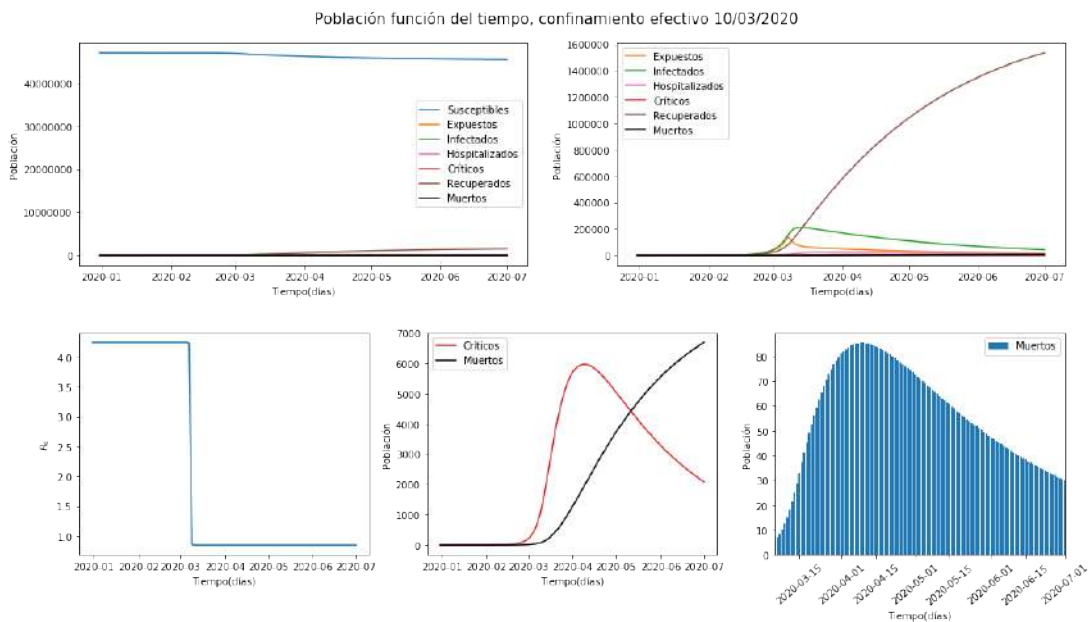
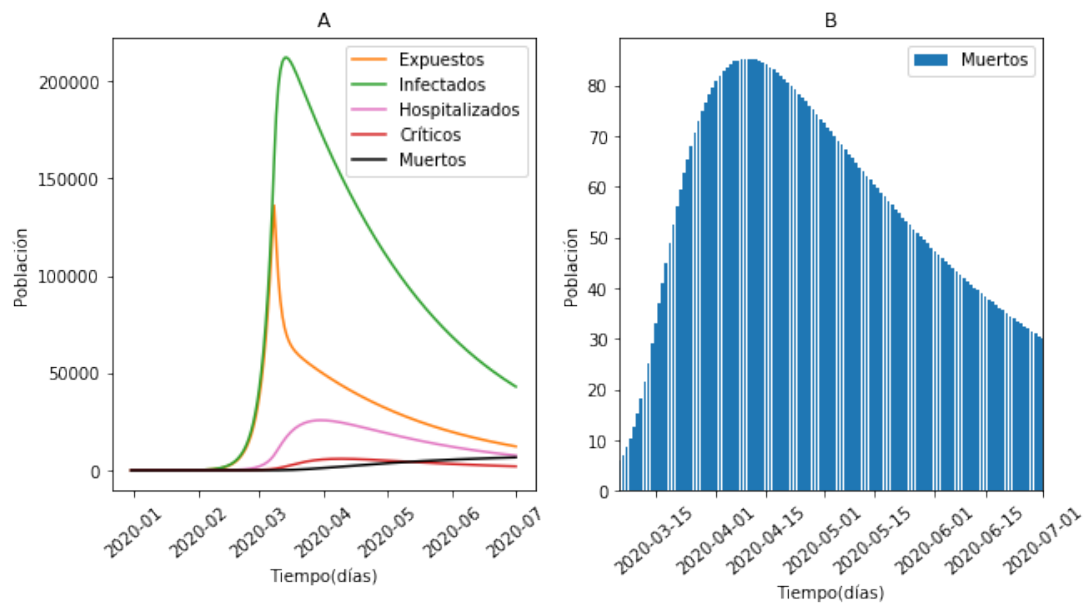


```

plt.figure(figsize = (18,9))
plt.suptitle("Población función del tiempo, confinamiento efectivo 10/03/2020",
    ↳ fontsize = 15)
plt.subplots_adjust(wspace = 0.2, hspace = 0.3, top = 0.93)
plt.subplot(2,2,1)
plt.plot(fechas, susceptibles, "-", label = "Susceptibles")
plt.plot(fechas, expuestos, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color =
    ↳ '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.subplot(2,3,4)
plt.plot(fechas, R_0t)
plt.xlabel("Tiempo(días)")
plt.ylabel("$R_0$")
plt.subplot(2,3,5)
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.subplot(2,3,6)
plt.bar(fechas, muertes_diarias, label = "Muertos")
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xlim(fechas[65], fechas[rango-1])
plt.xticks(rotation=40)
plt.legend()
plt.subplot(2,2,2)
plt.plot(fechas, expuestos, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color =
    ↳ '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')

```

```
plt.savefig('globalconf1003.png', dpi = 300, bbox_inches = 'tight', pad_inches = 0)
plt.show()
```



```
[24]: panorama=(susceptibles[123],  
↳exposeados[123],infectados[123],hospitalizados[123],  
↳criticos[123],recuperados[123],muertos[123])  
panorama, sum(panorama)
```

```
[24]: ((45806429.480833136, 31093.016318891758, 107740.78897839977, 18744.836923211704, 4991.5801135919, 105
```

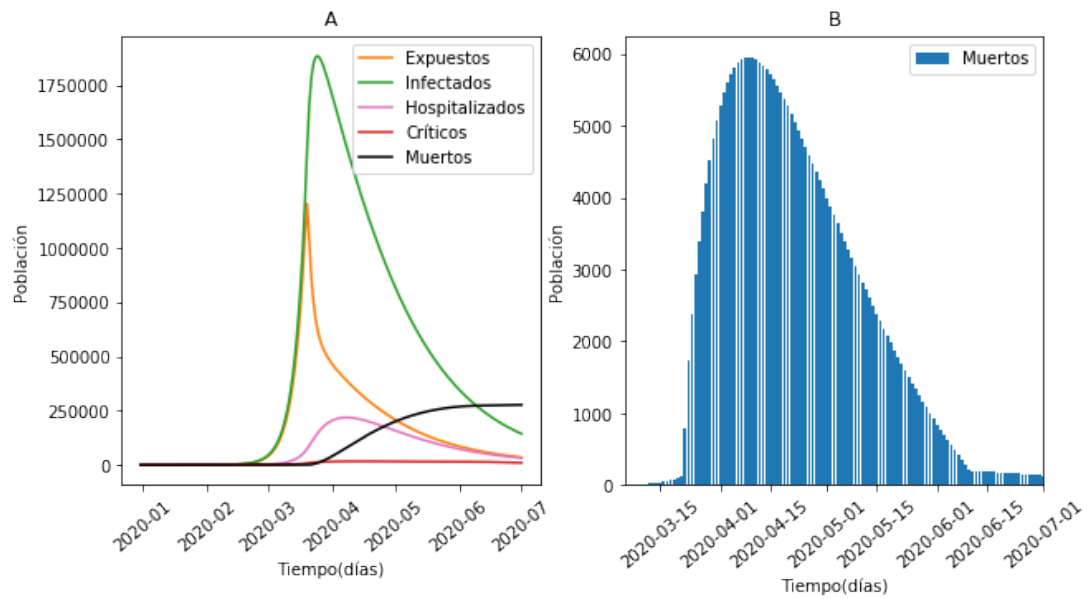
```
[25]: mod = modelo(rango, grupos, UCI_100, R_Oi, R_Of, k, 80, u, m, a, s)  
tiempo = np.linspace(0,rango,dias)  
susceptibles = mod[0]  
exposeados = mod[1]  
infectados = mod[2]  
hospitalizados = mod[3]  
criticos = mod[4]  
recuperados = mod[5]  
muertos= mod[6]  
R_Ot = mod[7]  
muertes_diarias=[0]  
for i in range(0, len(muertos)-1):  
    muertes_diarias.append(muertos[i+1]-muertos[i])  
plt.figure(figsize=(10,5))  
plt.subplot(1,2,1)  
plt.title("A")  
plt.plot(fechas, exposeados, "-", label = "Expuestos", color = '#ff7f0e')  
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')  
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color =  
↳'#e377c2' )  
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')  
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')  
plt.xlabel("Tiempo(días)")  
plt.ylabel("Población")  
plt.xticks(rotation=40)  
plt.legend()  
plt.ticklabel_format(style='plain', axis = 'y')  
plt.subplot(1,2,2)  
plt.title("B")  
plt.bar(fechas, muertes_diarias, label = "Muertos")  
plt.xlabel("Tiempo(días)")  
plt.ylabel("Población")  
plt.xlim(fechas[65], fechas[rango-1])  
plt.xticks(rotation=40)  
plt.legend()  
plt.savefig('2403.png', dpi = 300, bbox_inches = 'tight',pad_inches = 0)  
plt.show()  
  
plt.figure(figsize = (18,9))
```

```

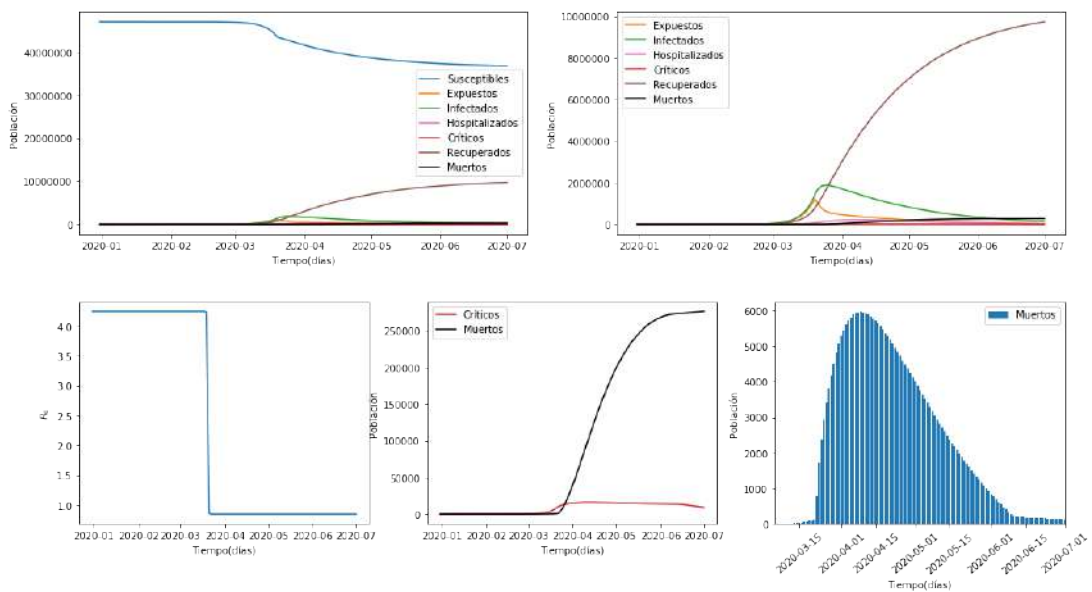
plt.suptitle("Población función del tiempo, confinamiento efectivo 24/03/2020",
    ↪ fontsize = 15)
plt.subplots_adjust(wspace = 0.2, hspace = 0.3, top = 0.93)
plt.subplot(2,2,1)
plt.plot(fechas, susceptibles, "-", label = "Susceptibles")
plt.plot(fechas, expuestos, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color =
    ↪ '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.subplot(2,3,4)
plt.plot(fechas, R_0t)
plt.xlabel("Tiempo(días)")
plt.ylabel("$R_0$")
plt.subplot(2,3,5)
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.subplot(2,3,6)
plt.bar(fechas, muertes_diarias, label = "Muertos")
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xlim(fechas[65], fechas[rango-1])
plt.xticks(rotation=40)
plt.legend()
plt.subplot(2,2,2)
plt.plot(fechas, expuestos, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color =
    ↪ '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.savefig('globalconf2403.png', dpi = 300, bbox_inches = 'tight', pad_inches =
    ↪ 0)

```

```
plt.show()
```



Población función del tiempo, confinamiento efectivo 24/03/2020



```
[26]: panorama=(susceptibles[123],  
               ↳exposeados[123],infectados[123],hospitalizados[123],  
               ↳criticos[123],recuperados[123],muertos[123])  
panorama, sum(panorama)
```

[26]: ((38552238.95307962, 203492.941990225, 803650.5111793799, 155297.8985465236, 15232.616089282432, 70937

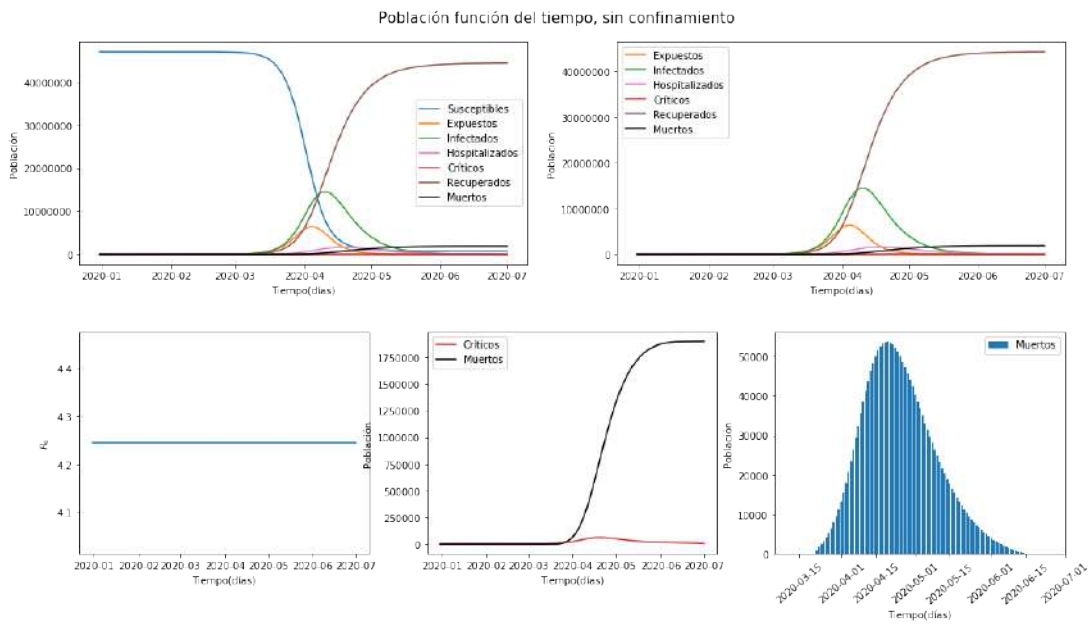
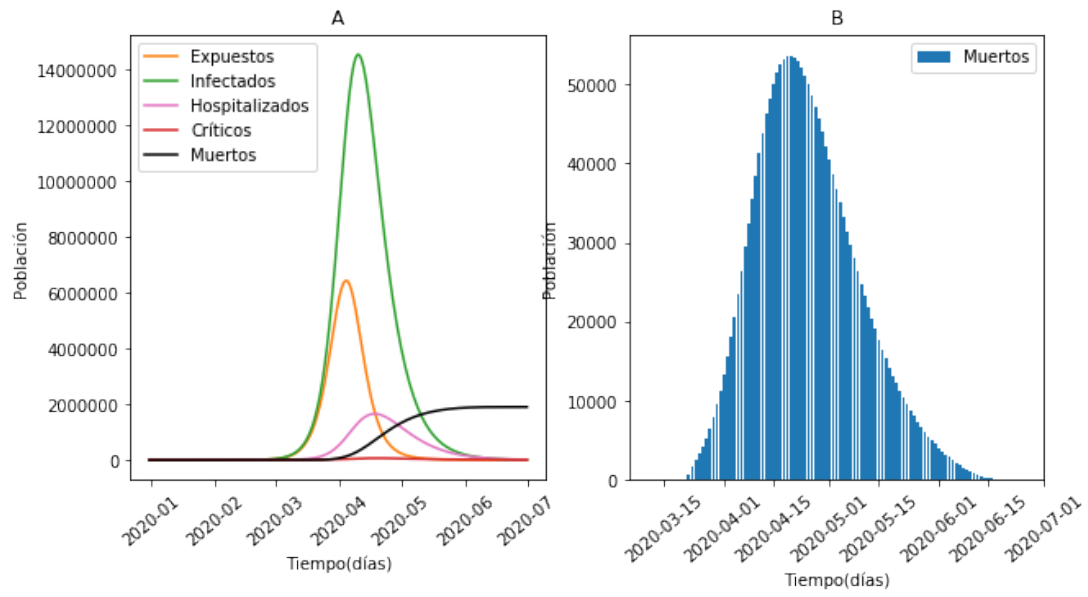
```
[27]: mod = modelo(rango, grupos, UCI_100, R_Oi, R_Of, k, 1000, u, m, a, s)
tiempo = np.linspace(0,rango,dias)
susceptibles = mod[0]
exposeados = mod[1]
infectados = mod[2]
hospitalizados = mod[3]
criticos = mod[4]
recuperados = mod[5]
muertos= mod[6]
R_Ot = mod[7]
muertes_diarias=[0]
for i in range(0, len(muertos)-1):
    muertes_diarias.append(muertos[i+1]-muertos[i])
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.title("A")
plt.plot(fechas, exposeados, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color = '#e377c2')
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xticks(rotation=40)
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.subplot(1,2,2)
plt.title("B")
plt.bar(fechas, muertes_diarias, label = "Muertos")
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xlim(fechas[65], fechas[rango-1])
plt.xticks(rotation=40)
plt.legend()
plt.savefig('NAN.png', dpi = 300, bbox_inches = 'tight',pad_inches = 0)
plt.show()

plt.figure(figsize = (18,9))
plt.suptitle("Población función del tiempo, sin confinamiento", fontsize = 15)
plt.subplots_adjust(wspace = 0.2, hspace = 0.3,top = 0.93)
plt.subplot(2,2,1)
plt.plot(fechas, susceptibles, "-", label = "Susceptibles")
plt.plot(fechas, exposeados, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
```

```

plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color = '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.subplot(2,3,4)
plt.plot(fechas, R_0t)
plt.xlabel("Tiempo(días)")
plt.ylabel("$R_0$")
plt.subplot(2,3,5)
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.subplot(2,3,6)
plt.bar(fechas, muertes_diarias, label = "Muertos")
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.xlim(fechas[65], fechas[rango-1])
plt.xticks(rotation=40)
plt.legend()
plt.subplot(2,2,2)
plt.plot(fechas, exposeados, "-", label = "Expuestos", color = '#ff7f0e')
plt.plot(fechas, infectados, "-", label = "Infectados", color = '#2ca02c')
plt.plot(fechas, hospitalizados, "-", label = "Hospitalizados", color = '#e377c2' )
plt.plot(fechas, criticos, label = "Críticos", color = '#d62728')
plt.plot(fechas, recuperados, label = "Recuperados", color = '#8c564d')
plt.plot(fechas, muertos, "-", label = "Muertos", color = 'black')
plt.xlabel("Tiempo(días)")
plt.ylabel("Población")
plt.legend()
plt.ticklabel_format(style='plain', axis = 'y')
plt.savefig('globalconfNAN.png', dpi = 300, bbox_inches = 'tight', pad_inches = 0)
plt.show()

```



```
[28]: panorama=(susceptibles[123],  
↳exposeados[123],infectados[123],hospitalizados[123],  
↳criticos[123],recuperados[123],muertos[123])  
panorama, sum(panorama)
```

```
[28]: ((1052540.9883135231, 203880.169308637, 3675727.7113151643, 1122122.2483826855, 49151.337178211696, 39
```