



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

SIMULACIÓN Y CLASIFICACIÓN DE LA MOVILIDAD EN UNA
CIUDAD DURANTE UNA PANDEMIA UTILIZANDO MODELACIÓN
BASADA EN AGENTES Y RECONOCIMIENTO DE ACTIVIDAD
HUMANA

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
JUAN URIEL LEGARIA PEÑA

DIRECTOR DE TESIS:
DR. ERIK MOLINO MINERO RE
IIMAS, UNAM

CIUDAD UNIVERSITARIA, CD. MX. SEPTIEMBRE, 2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice general

1	Agradecimientos	I
2	Resumen	II
3	Introducción	IV
§3.1	Planteamiento del problema	V
§3.2	Objetivos	V
§3.3	Objetivo general	V
§3.4	Hipótesis	VII
§3.5	Estructura de tesis	VII
4	Marco teórico	1
§4.1	Modelos epidemiológicos SEIRD	1
§4.1.1	Incorporación de comportamiento periódico en el modelo compartimental SEIRD	3
§4.2	Modelado de epidemias bajo una perspectiva de sistemas complejos	6
§4.2.1	Autómatas celulares	7
§4.2.2	Redes	8
§4.2.3	Modelos basados en agentes	9
§4.3	Algunos trabajos y modelos existentes para simulación de pandemias	14
§4.3.1	COVID - ABS	15
§4.3.2	INFEKTA	16
§4.3.3	Relación entre uso de mascarilla, transmisión asintomática y distanciamiento social en la propagación de COVID - 19	17
§4.3.4	Distanciamiento social y resurgimiento de epidemias en modelos basados en agentes SIR	18
§4.4	Reconocimiento de actividad humana	19
§4.4.1	Características del dominio de la frecuencia y Wavelet	22
§4.4.2	Modelos de clasificación	29

5	Metodología	35
§5.1	Modelo basado en agentes para simular movilidad de personas durante una pandemia	37
§5.1.1	Entorno virtual	38
§5.1.2	Módulos de comportamiento de los agentes	40
§5.1.3	Módulo de salud	41
§5.1.4	Módulo de actividades	46
§5.1.5	Módulo de movilidad	51
§5.1.6	Módulo de protección y políticas implementadas	52
§5.2	Clasificación del estado de salud a partir de datos de movilidad	57
6	Calibración y ajuste de parámetros	65
§6.1	Modelo basado en agentes	65
§6.1.1	Ajuste de parámetros epidemiológicos	65
§6.1.2	Calibración de las actividades de los agentes utilizando una encuesta	67
§6.1.3	Calibración del número de agentes	72
§6.1.4	Ajuste de parámetros en las políticas de protección y resguardo	72
§6.2	Clasificación de datos de movilidad	75
§6.2.1	Parámetros utilizados en la segmentación de series de tiempo y extracción de características	75
§6.2.2	Parámetros utilizados en los clasificadores	76
7	Resultados y discusión	77
§7.1	Modelo basado en agentes	77
§7.1.1	Series de tiempo SEIRD	77
§7.1.2	Series de tiempo de asistencia a distintos lugares de la ciudad	79
§7.1.3	Comparación de las distintas políticas de resguardo y protección implementadas	82
§7.2	Series SEIRD al incorporar nuevas variantes.	85
§7.3	Clasificación de datos de movilidad	87
§7.3.1	Evaluación de los distintos clasificadores entrenados y comparación de la predicción al aplicar políticas de resguardo	87
8	Conclusión	91
§8.0.1	Conclusiones	91
§8.0.2	Trabajo a futuro	92

Índice de figuras

4.1	Gráfico de una de las soluciones que se obtienen al resolver el sistema de ecuaciones SEIRD. Los valores utilizados para los parámetros fueron $\alpha = 10$, $\beta = 2$, $\gamma_1 = 1$, $\gamma_2 = 0.5$, y los de las condiciones iniciales $S_0 = 0.99$, $E_0 = 0$, $I_0 = 0.01$, $R_0 = 0$, $D_0 = 0$	3
4.2	Parámetros importantes que pueden evaluarse en curvas SEIRD. t_p denota el tiempo que toma llegar al pico de infección, h_p es la altura de dicho pico, D_∞ es el número de fallecidos al culminar la epidemia y $FWHM$ es la anchura a media altura.	4
4.3	Ejemplo de soluciones cíclicas obtenidas al introducir el término no lineal propuesto para generar olas en las ecuaciones SEIRD. Los valores utilizados para los parámetros fueron $\alpha = 0.94$, $\beta = 0.3$, $\gamma_1 = 0.2$, $\gamma_2 = 0$, $\mu = 5$, $C_{\text{sat}} = 2$ y las condiciones iniciales $S_0 = 0.99$, $E_0 = 0$, $I_0 = 0.01$, $R_0 = 0$, $D_0 = 0$	5
4.4	Propagación de una pandemia en un autómata celular de dimensión 100×100 en tres instantes de tiempo distintos $t = 0$, $t = 150$ y $t = 500$. Los valores utilizados para las probabilidades de transición fueron $\alpha = 0.9$, $\beta = 0.05$, $\gamma_1 = 0.01$ y $\gamma_2 = 0.005$	8
4.5	Evolución de una pandemia en una una red aleatoria de 100 individuos en tres instantes de tiempo distintos: $t = 0$, $t = 150$ y $t = 200$. Los valores utilizados para las probabilidades de transición fueron $\alpha = 0.9$, $\beta = 0.05$, $\gamma_1 = 0.01$, $\gamma_2 = 0.005$	9
4.6	Evolución de una pandemia utilizando un modelo basado en agentes donde estos se mueven aleatoriamente en un espacio rectangular. El número de agentes utilizado fue 500, y los parámetros de infección fueron $\alpha = 0.9$, $\beta = 0.05$, $\gamma_1 = 0.01$ y $\gamma_2 = 0.005$	14
4.7	En HAR las series de tiempo de actividad de una persona se dividen en ventanas, y se asigna a cada una de ellas una etiqueta, que indica el estado o actividad de la persona en ese segmento de tiempo.	20
4.8	Flujo de trabajo del reconocimiento de actividad humana.	21

ÍNDICE DE FIGURAS

4.9	Serie de tiempo de la frecuencia promedio de las señales de posición de un agente. Las secciones en rojo corresponden a lapsos en que el individuo se encontraba enfermo. La unidad 1/día representa el número de veces por día que oscila la señal del agente en el respectivo eje.	25
4.10	Cascada de filtros utilizada para descomponer una señal en su aproximación y sus señales de detalle en el análisis multiresolución con wavelets.	28
4.11	Series de tiempo de las energías de la aproximación Wavelet y 3 detalles de la señal de movimiento horizontal $x[t]$ de un agente. Los segmentos en color rojo corresponden a periodos donde el agente se encontró enfermo.	29
5.1	Esquema general de la metodología desarrollada	35
5.2	Mundo virtual donde se desarrolla la simulación	38
5.3	Módulos de comportamiento de los agentes	40
5.4	El módulo de guardado de datos puede pensarse como una cámara o sensor que sigue al agente en todo momento y recopila datos de su actividad, como son sus coordenadas de posición y su estado de salud.	41
5.5	Transiciones del estado de salud de los agentes.	42
5.6	Regla de interacción para la selección y ejecución de actividades	48
5.7	Variación temporal de la probabilidad e incremento de probabilidad para dos actividades distintas: trabajar y ejercitarse.	50
5.8	Regla de interacción para el movimiento de los agentes dentro de la ciudad.	53
5.9	Efecto de las políticas implementados y los parámetros del modelo que se ven afectados por ellas.	54
5.10	Pasos que se siguieron en la clasificación de los datos de movilidad.	58
6.1	Distribuciones de edad, estado de residencia y género de los participantes que contestaron la encuesta utilizada para calibrar las actividades de los agentes.	68
6.2	Distribuciones de frecuencia, duración y duración de las distintas actividades consideradas en el modelo de agentes, construidas usando información del cuestionario aplicado.	71
6.3	Distribuciones de uso de cubrebocas en los distintos tipos de lugar considerados en el modelo de agentes, construidas con información de una encuesta.	73
6.4	Distribución de vacunas aplicadas en México (Información obtenida del 28 de Septiembre al 10 de Octubre del año 2021).	74

ÍNDICE DE FIGURAS

6.5	Distribuciones de la influencia que tiene el semáforo epidemiológico en los individuos encuestados.	74
6.6	Madre Wavelet Daubechies - 4 utilizada en la extracción de características en el dominio Wavelet.	76
7.1	Gráficos SEIRD obtenidos en las distintas políticas de resguardo y protección implementadas.	78
7.2	Series de tiempo de asistencia a los lugares de la ciudad simulada en el caso donde no se aplicó ninguna política de protección y resguardo.	80
7.3	Series de tiempo de asistencia a los lugares de la ciudad simulada en el caso donde se aplicó una política de aislamiento. Los segmentos encuadrados en verde corresponden a lapsos de tiempo donde se encendieron las restricciones de movimiento debido a que se superó un umbral en la ocupación hospitalaria.	81
7.4	Comparación de distintas métricas obtenidas para cada uno de los escenarios de protección simulados. En la figura N = Escenario nulo (sin política), M = Uso de mascarilla, V = Vacunación, A = aislamiento, DS = Distanciamiento Social y S = Semáforo.	83
7.5	Valor del parámetro propuesto presentado en la ecuación 7.1 para distintas políticas. En la figura N = Escenario nulo (sin política), M = Uso de mascarilla, V = Vacunación, A = aislamiento, DS = Distanciamiento Social y S = Semáforo.	84
7.6	Gráficos SEIRD de los escenarios sin política y aislamiento considerando aparición de nuevas variantes del patógeno.	86
7.7	Curvas ROC y PR de los distintos clasificadores. Los datos de movilidad individual utilizados para entrenar fueron generados en un escenario sin política.	88
7.8	Curvas ROC y PR de los distintos clasificadores. Los datos de movilidad individual utilizados para entrenar fueron generados en un escenario de aislamiento.	89
8.1	Propuesta de uso de un algoritmo de hormigas para entrenar rutas de un lugar a otro dentro de la ciudad simulada.	93

Índice de Algoritmos

4.1	Regla de interacción para el movimiento aleatorio de un agente (Modelo ilustrativo de agentes).	12
4.2	Regla de interacción para actualizar el estado de salud del agente (Modelo ilustrativo de agentes).	13
4.3	Cálculo de la transformada discreta de Fourier	23
5.1	Algoritmo de actualización de día y hora dentro del mundo simulado.	39
5.2	Algoritmo de obtención del vecindario de los agentes.	44
5.3	Regla de interacción para actualizar el estado de salud en la simulación.	45
5.4	Regla de interacción para actualizar la probabilidad de las actividades.	51

Índice de tablas

5.1	Eficiencias a utilizar en la política de uso de mascarilla.	54
6.1	Valor asignado a los parámetros epidemiológicos utilizados dentro del modelo basado en agentes desarrollado.	66
6.2	Parámetros utilizados para entrenar los distintos clasificadores	76
7.1	Métricas de evaluación de los distintos clasificadores para un caso donde se aplicó ninguna política de resguardo.	87
7.2	Métricas de evaluación de los distintos clasificadores para un caso donde se aplicó una política de aislamiento.	89

Agradecimientos

Agradezco a mi tutor, el Dr. Erik Molino Minero por guiarme en la elaboración de esta tesis. Aprendí mucho de usted y su punto de vista me mostró lo creativo e interesante que puede ser plantear una investigación utilizando herramientas computacionales.

También doy gracias a los profesores de IIMAS Mérida quienes a pesar de las condiciones de pandemia siempre nos dieron la mejor ayuda para comprender las asignaturas que nos impartieron.

Agradezco a los investigadores que me permitieron colaborar o presentar con ellos algún proyecto, como fueron la Dra. Nidiyare Hevia Montiel, el Dr. Jorge Luis Perez Gonzalez, la Dra. Yuriria Cortés Poza y el Dr. Jose Antonio Neme Castillo. He de agradecer a Antonio también las clases que imparte, que son de las más claras e interesantes que he cursado. Espero poder leer en el futuro todas las obras que nos recomendó a lo largo de sus cursos.

A mis padres y abuelos quienes siempre me han apoyado y representan el pilar fundamental de todos mis logros.

A mis amigos Félix, Gemma, Alejandro, Paco y Fer, a quienes aprecio mucho y a mis compañeros de IIMAS Mérida quienes a pesar de haber conocido en línea, hicieron muy amenos los cursos que tomamos.

Finalmente agradezco a CONACYT por el apoyo económico que me proporcionaron a lo largo de mis estudios.

Resumen

Resumen

En este trabajo se implementó un modelo de agentes para simular el movimiento de personas en una ciudad donde ocurre una pandemia. Dicha simulación fue utilizada para generar datos GPS sintéticos de los individuos, con el objetivo de investigar si era posible predecir el estado de salud de una persona usando sus datos de movilidad. El modelo propuesto incluye funcionalidades no contempladas en simulaciones de agentes existentes para COVID - 19, como son la generación de nuevas variantes del virus y la posibilidad de tener múltiples olas de infección. También abarca distintas políticas de protección y resguardo como son uso de mascarilla, aislamiento, vacunación, distanciamiento social y una combinación de varias de ellas. Una comparación entre las curvas de infección, fallecimientos y ocupación hospitalaria para las distintas políticas mostró que la estrategia más adecuada de control es una combinación de aislamiento, uso de mascarilla y distanciamiento social, aunque por si solo el uso de mascarilla provee una medida adecuada, que logra extender el periodo de pandemia y disminuir la altura del pico de infección. Para evaluar la viabilidad de predecir el estado de salud de personas a partir de su movimiento se aplicaron técnicas de reconocimiento de actividad humana (HAR). En primer lugar, las series sintéticas GPS se dividieron en ventanas y para cada segmento se obtuvieron distintas características en los dominios de la frecuencia y wavelet. Los vectores de características extraídos para cada ventana se utilizaron posteriormente para entrenar y evaluar distintos modelos de clasificación. En los resultados de dicha evaluación, los bosques aleatorios y los ensambles de clasificación consiguieron las mejores métricas. Así mismo, se encontró que aplicar una política de aislamiento resultaba en un peor desempeño de los clasificadores, lo cual implica que al aplicar dicha intervención se pierde parte de la relación entre movilidad y transmisión del virus.

Abstract

In this work we set out to implement an agent based model, to simulate the movement of persons living in a city where a pandemic takes place. The simulation was used to generate synthetic GPS data of individuals, to research if it is possible to predict the health state of a person based on its movement patterns. The proposed model includes functionalities not yet contemplated in available COVID-19 simulations, such as the emergence of new virus variants and simulation of multiple waves of infection. It also supports several policies and interventions that can be applied to a population, such as mask usage, isolation, vaccination, social distancing and a combination of some of them. A comparison of infection, death, and hospital attendance curves across the different scenarios showed that the most appropriate strategy for virus transmission control is a combination of isolation, social distancing and mask use, however, mask use alone is able to extend the infection curve for a significant amount of time, and lower the infection peak. To evaluate the viability of predicting the health state of an individual using their movement data, methods of Human Activity Recognition (HAR) were applied. First the synthetic GPS series were divided in segments and for each window a series of features in the frequency and wavelet domains were extracted. The feature vectors extracted for each window were then used to train and evaluate several classification models. Results showed that the best classification metrics in this problem are obtained for a classifier ensemble and random forests. Also, when the mobility data was generated while applying an isolation policy, the classifiers had more difficulty predicting the health state of individuals, showing that a correlation between mobility and transmission of the virus is lost when this intervention is applied.

Introducción

Durante la pandemia de COVID-19, parte de los esfuerzos de investigación fueron dirigidos a determinar los procesos ligados a la propagación del virus, y a establecer medidas adecuadas para controlarla.

Al respecto se hizo evidente que la movilidad de la población constituía uno de los factores cruciales a monitorear y regular para contener la diseminación de SARS-CoV-2, y por tal razón varias de las políticas instauradas se centraron en disminuir las salidas de los habitantes dentro de las ciudades.

Además de la aplicación de políticas de resguardo, se estudió la forma en que se podía utilizar la información de movilidad para alertar a individuos sobre posibles contagios de acuerdo a los lugares que hubiesen visitado. Así, surgieron las llamadas aplicaciones de rastreo de contactos, las cuales registraban aquellos usuarios que se hubieran encontrado cerca en un momento determinado, alertándolos en caso de que alguno de ellos llegara a reportar síntomas de COVID - 19 [1].

Una de las desventajas asociadas a dichas aplicaciones, sin embargo, es que su efectividad requería que como mínimo 39.5% a 43.6% de la población las utilizara simultáneamente [2], lo cual es un porcentaje inviable a conseguir en ciertas comunidades.

Ante ello, una alternativa que pudiese resultar efectiva, y que no requeriría de un uso simultáneo por parte de un gran número de usuarios, es el uso de técnicas de reconocimiento de actividad humana. En ellas, lo que se buscaría es tomar segmentos cortos de las coordenadas GPS de una persona, e indicar con base en ellas el estado de salud del individuo aplicando clasificadores entrenados, como los que suelen utilizarse en el aprendizaje de máquina.

La principal dificultad de esta propuesta yace en hallar una base de datos apropiada, ya que para entrenar el clasificador mencionado se requerirían datos de posición de personas y de su estado de salud, lo cual no se suele reportar por motivos de privacidad y seguridad. En efecto, aquellas compañías que ofrecen servicios GPS como Google, Facebook o Apple,

suelen publicar únicamente variables que surgen de un comportamiento colectivo de muchas personas como son la asistencia a distintos tipos de lugar (parques, oficinas, etc.) o mapas de rango de movimiento de los habitantes [3, 4, 5].

En esta tesis se propone diseñar e implementar un modelo basado en agentes, que simule el movimiento de individuos en una ciudad donde se desarrolla una pandemia. Los datos generados con dicho modelo se aplicarán posteriormente para indagar, al menos en forma aproximada, si es posible predecir el estado de salud de una persona usando sus datos de movimiento, aplicando técnicas de reconocimiento de actividad humana.

Algo importante a mencionar es que la intención no necesariamente es extrapolar los resultados generados a un escenario real, ya que a pesar de los múltiples elementos contemplados en el modelo, sería incorrecto afirmar que se está replicando en forma fiel el comportamiento de una población en concreto. El interés es investigar en un experimento virtual controlado la relación entre movilidad y estado de salud, explorando si ciertas herramientas de inteligencia artificial son capaces de abstraer los patrones de movimiento de una persona, a tal grado que puedan detectar cuando esta se halla enferma.

3.1. Planteamiento del problema

El problema a resolver en este trabajo es la simulación de actividad y movimiento de personas en una ciudad donde se transmite un virus, y la posterior clasificación de dichos datos para predecir el estado de salud de los individuos usando series de tiempo de sus coordenadas.

Otro problema contemplado dentro de la simulación de datos es la implementación de distintas políticas e intervenciones, como son uso de cubrebocas, aislamiento, distanciamiento social y vacunación, para estudiar si tales medidas afectan la predicción del estado de salud usando datos GPS, y además realizar una comparación entre ellas, examinando cuales consiguen un mejor control de la pandemia.

3.2. Objetivos

3.3. Objetivo general

El objetivo general de esta tesis es proponer e implementar un modelo basado en agentes, que simule el movimiento de individuos en una ciudad donde se transmite una enfermedad. Dentro de tal modelo será importante dilucidar como modelar en forma ade-

cuada políticas como uso de cubrebocas, aislamiento, distanciamiento social y vacunación, además de elementos adicionales como la aparición de nuevas variantes del patógeno. Una vez cumplido tal cometido, se buscará aplicar técnicas de reconocimiento de actividad humana, para predecir el estado de salud de personas a partir de sus datos GPS sintéticos.

Las políticas modeladas servirán concretamente para hacer una comparación de los escenarios, y determinar cual(es) de ellos consiguen un control más eficaz de la transmisión del virus. Además, se examinará si la imposición de políticas que afectan la movilidad hacen que la predicción del estado de salud se vuelva una tarea más complicada, como sugieren ciertos estudios [6].

Objetivos particulares

La secuencia de metas que se han planteado para conseguir el objetivo general antes descrito son las que se desglosan a continuación:

1. Estudiar en la literatura los modelos existentes para simular pandemias.
2. Plantear un modelo de agentes en el que se simulen las actividades y movimiento de personas en una ciudad, considerando un patógeno basado en COVID-19 que se va transmitiendo entre ellas.
3. Incorporar al modelo ciertas políticas e intervenciones que se han propuesto ante la pandemia actual, como son uso de cubrebocas, aislamiento, distanciamiento social, vacunación y escenarios donde varias de ellas se combinen.
4. Calibrar la conducta de los agentes utilizando información de una encuesta aplicada en México. Específicamente, se consultarán la frecuencia, lugares de realización y duración de las distintas actividades que se han incluido como parte de la rutina cotidiana de los agentes.
5. Exportar series de tiempo de las fracciones de población susceptible, expuesta, infectada, recuperada, fallecida y de la ocupación hospitalaria para cada una de las políticas, y con ellas hacer una comparación de la calidad en el control de la pandemia que se consigue en cada escenario.
6. Exportar series de tiempo de la posición y estado de salud de varios agentes en un escenario sin política y entrenar con ellas varios tipos de clasificadores (bosques aleatorios, SVM, red neuronal, ensamble). Al final del entrenamiento, dichos clasificadores serán capaces de arrojar una etiqueta del estado de salud de una persona en

base a un segmento corto de sus datos de posición. Los clasificadores serán evaluados usando distintas métricas como accuracy, precision, recall, entre otras.

7. Repetir el objetivo anterior con datos generados en un escenario de aislamiento, para examinar si la aplicación de medidas de resguardo afecta en la calidad de predicción de los clasificadores.

3.4. Hipótesis

La hipótesis que se tiene sobre la comparación de intervenciones es que las que conseguirán un mejor control de la pandemia serán la vacunación y el uso combinado de varias de ellas. En el primer caso, nuestra sospecha surge de que la vacunación disminuye directamente tanto la probabilidad de contagio como la posibilidad de presentar un caso grave de la enfermedad, lo cual puede tener un efecto importante en el tamaño de pico de infección y la fracción final de fallecidos. Por otra parte, aplicar varias medidas al mismo tiempo podría tener un efecto sinérgico considerable, dada la naturaleza no lineal del fenómeno que se está intentando regular.

En la clasificación de datos de movilidad nuestra visión es que si se logrará hacer una buena predicción del estado de salud de las personas, dadas las simplificaciones y simetrías que presenta el comportamiento de los individuos en nuestra población idealizada. En dicha tarea, se prevé que aquellos clasificadores que son de uso general, como los árboles aleatorios o las máquinas de soporte vectorial, conseguirán adecuadas métricas de desempeño. Finalmente, se anticipa que al aplicar medidas como el aislamiento la tarea de predicción del estado de salud se complicará, ya que esto introduce nuevas modalidades de comportamiento en los habitantes, que pueden afectar a la relación entre los procesos de movilidad y propagación del virus.

3.5. Estructura de tesis

Esta tesis dará comienzo con un marco teórico (Capítulo 4), en el que se introducen aquellas ideas y conceptos relativos al modelado de pandemias y reconocimiento de actividad humana. Una vez sentada dicha base, se desglosará en el Capítulo 5 la estructura del modelo de agentes propuesto y la metodología usada para clasificar los datos de movilidad simulados. La calibración de los parámetros del modelo de agentes y la encuesta de actividades que se utilizó para ajustar la conducta de la población virtual se explicará en el capítulo 6, y una vez hecho esto se procederá a enunciar y discutir los resultados obtenidos

en el Capítulo 7. Finalmente, se cerrará con el trabajo en el capítulo 8, haciendo una recapitulación final de lo obtenido y contrastándolo con los objetivos planteados inicialmente. Así mismo, se ofrecerá una tentativa visión de los futuros caminos que podrían seguirse para continuar y mejorar esta investigación.

Marco teórico

En esta sección se describen los antecedentes y conceptos teóricos que ayudaron a establecer la metodología de este trabajo. Se comienza introduciendo algunas técnicas comúnmente utilizadas para modelar sistemas complejos, enfocando la narración a entender cómo se les puede aplicar para simular la evolución de una epidemia. Lo anterior sustentará el modelo basado en agentes que se desarrolló para simular el movimiento de personas durante una pandemia, y proporcionará a su vez una revisión de modelos relacionados que pueden encontrarse en la literatura. Hecho esto, en la parte final del capítulo se dará una introducción al reconocimiento de actividad humana y a las técnicas de clasificación utilizadas en dicha área. Tales herramientas se aplicaron dentro de la metodología para estudiar si era posible conocer el estado de salud de personas a partir de sus datos de movimiento.

4.1. Modelos epidemiológicos SEIRD

Un primer paso a dar al estudiar un sistema o fenómeno, es seleccionar las variables que se utilizarán para describirlo. Dicha decisión resulta importante, porque determina el tipo de información con la que se puede contar para hacer análisis y/o tomar decisiones.

En el caso particular de una pandemia, que es el fenómeno de interés en esta tesis, las variables macroscópicas que suelen estudiarse son las fracciones de población que se hallan en distintos estados de salud. Así, se suele comúnmente seguir la evolución de la fracción de personas susceptibles a enfermarse $S(t)$, la fracción de infectados $I(t)$, la fracción de recuperados $R(t)$, y la fracción de fallecidos $D(t)$. En enfermedades con un periodo de incubación significativo pueden incluso considerarse los expuestos $E(t)$, que representan personas ya contagiadas que aun no presentan cambios fisiológicos debidos a la enfermedad.

Los cambios en el tiempo que sufren estas variables pueden modelarse de diversas formas. En esta sección se explicará en particular el modelo compartimental SEIRD, en el

que su dinámica se plantea utilizando ecuaciones diferenciales. A pesar de que esta no es la forma en la que se les obtendrá en esta tesis, representa un punto de comparación importante, ya que cómo se verá más adelante, los resultados que arroja nuestro modelo para estas variables se asemejan al tipo de soluciones que se obtienen al resolver las ecuaciones de SEIRD. En el modelo propuesto $S(t)$, $E(t)$, $I(t)$, $R(t)$ y $D(t)$ se obtendrán contando explícitamente el número de personas del mundo simulado que se hallan en cada uno de los estados de salud, y dividiendo las cuentas entre el total de población para obtener fracciones.

El sistema de ecuaciones diferenciales a resolver en el modelo SEIRD es el que sigue:

$$\bullet \frac{dS}{dt} = -\alpha S(t)I(t) \quad (4.1)$$

$$\bullet \frac{dE}{dt} = \alpha S(t)I(t) - \beta E(t) \quad (4.2)$$

$$\bullet \frac{dI}{dt} = \beta E(t) - (\gamma_1 + \gamma_2)I(t) \quad (4.3)$$

$$\bullet \frac{dR}{dt} = \gamma_1 I(t) \quad (4.4)$$

$$\bullet \frac{dD}{dt} = \gamma_2 I(t) \quad (4.5)$$

En dichas ecuaciones α representa la tasa con la que los susceptibles se contagian al interactuar con los infectados, β representa la tasa con que los expuestos pasan a infectados, γ_1 es la tasa con que los infectados se recuperan y γ_2 es la tasa con que los infectados fallecen.

Una propiedad asociada a dichas ecuaciones, es que su suma es igual a cero, lo cual implica que la suma de $S(t)$, $E(t)$, $I(t)$, $R(t)$ y $D(t)$ permanece constante:

$$\frac{d(S(t) + E(t) + I(t) + R(t) + D(t))}{dt} = 0 \implies S(t) + E(t) + I(t) + R(t) + D(t) = \text{cte} \quad (4.6)$$

La expresión anterior puede interpretarse como una ley de conservación, la cual afirma que el tamaño de la población de estudio permanece constante en el tiempo. Además de ello la propiedad impone una restricción geométrica importante en el tipo de soluciones que se pueden generar. En el caso en que se trabaja con fracciones de población, la suma de todas las variables de estudio debiese ser igual a uno.

En la figura 4.1 se muestra el tipo de solución que se obtiene al resolver el sistema de ecuaciones antes planteado utilizando un integrador numérico Runge - Kutta de orden 4.

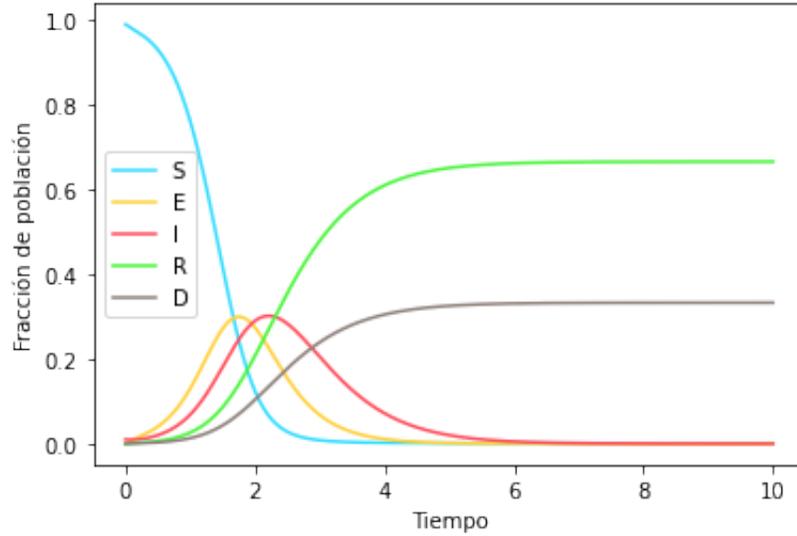


Figura 4.1: Gráfico de una de las soluciones que se obtienen al resolver el sistema de ecuaciones SEIRD. Los valores utilizados para los parámetros fueron $\alpha = 10$, $\beta = 2$, $\gamma_1 = 1$, $\gamma_2 = 0.5$, y los de las condiciones iniciales $S_0 = 0.99$, $E_0 = 0$, $I_0 = 0.01$, $R_0 = 0$, $D_0 = 0$.

El patrón que se observa en la figura como veremos, se asemeja a lo que arroja para dichas variables el modelo basado en agentes implementado en este trabajo. En este punto, vale la pena mencionar ciertos parámetros relevantes para evaluar la gravedad de una epidemia que pueden obtenerse usando las soluciones SEIRD. Estos son el tamaño de pico de infección h_p , el tiempo que toma llegar a dicho pico t_p , la fracción final de fallecidos D_∞ y la anchura a media altura $FWHM$, definida como el intervalo de tiempo entre los dos puntos de la curva de infección donde la altura es igual $\frac{h_p}{2}$. Una interpretación gráfica de tales parámetros se ha mostrado en la figura 4.2.

4.1.1. Incorporación de comportamiento periódico en el modelo compartimental SEIRD

Dentro del modelo desarrollado en esta tesis, uno de los fenómenos que se buscó incorporar fueron los patrones cíclicos de infección, también llamados olas, que suelen generarse en la naturaleza debido a las estaciones del año o la aparición de nuevas variantes de un patógeno.

Para modelar dicho oleaje, lo que se hizo primero fue estudiar como era posible introducirlo en las ecuaciones diferenciales SEIRD, para posteriormente trasladarlo al modelo de agentes que se implementó. Consultando en la literatura, en [7] se determinó que el uso

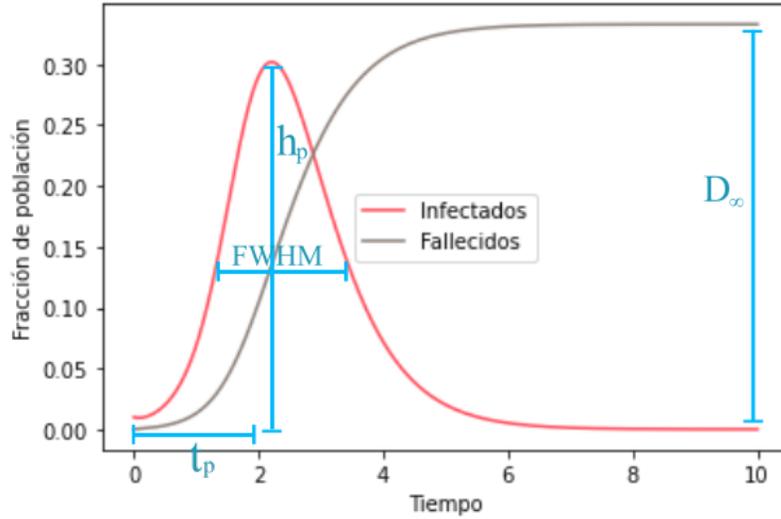


Figura 4.2: Parámetros importantes que pueden evaluarse en curvas SEIRD. t_p denota el tiempo que toma llegar al pico de infección, h_p es la altura de dicho pico, D_∞ es el número de fallecidos al culminar la epidemia y $FWHM$ es la anchura a media altura.

del término no lineal que se muestra en la ecuación 4.7 lograba que se generaran soluciones periódicas en las curvas SEIRD.

$$\frac{\mu S(t)I(t)^l}{C_{\text{sat}} + I^h} \quad (4.7)$$

Donde h y l representan exponentes constantes, C_{sat} es un parámetro conocido como constante de saturación y μ es la tasa con la que los susceptibles se vuelven infectados debido a este término.

En el trabajo citado esta expresión representaba una interacción entre infectados y susceptibles, en la que los susceptibles comenzaban a tomar precauciones conforme se informaban sobre la situación de pandemia que estaban viviendo. Para nuestros propósitos, sin embargo, se buscaba un oleaje debido a que las personas recuperadas volvían a ser susceptibles cuando aparecía una nueva variante del virus. Por ello, se modificó del término anterior, de tal forma que representará una interacción entre recuperados y susceptibles, en vez de infectados y susceptibles. La expresión final propuesta es la siguiente:

$$\mu \frac{S^2(t)R(t)}{C_{\text{sat}} + S(t)} \quad (4.8)$$

Donde se han hecho $h = 1$ y $l = 1$ en la ecuación 4.7 y se ha sustituido $S(t)$ por $R(t)$ e $I(t)$ por $S(t)$. Este término se sustrajo de la ecuación diferencial de los recuperados (Ecuación 4.4) y se añadió a la de susceptibles (Ecuación 4.1), dejando el sistema de

ecuaciones original como sigue:

$$\bullet \frac{dS}{dt} = -\alpha S(t)I(t) + \mu \frac{S^2(t)R(t)}{C_{\text{sat}} + S(t)} \quad (4.9)$$

$$\bullet \frac{dE}{dt} = \alpha S(t)I(t) - \beta E(t) \quad (4.10)$$

$$\bullet \frac{dI}{dt} = \beta E(t) - (\gamma_1 + \gamma_2)I(t) \quad (4.11)$$

$$\bullet \frac{dR}{dt} = \gamma_1 I(t) - \mu \frac{S^2(t)R(t)}{C_{\text{sat}} + S(t)} \quad (4.12)$$

$$\bullet \frac{dD}{dt} = \gamma_2 I(t) \quad (4.13)$$

Un detalle importante a mencionar es que previo a proponer el término anterior se probaron formas lineales de retroalimentación en el sistema. Estas, sin embargo, no conseguían el oleaje deseado, sino que inducían comportamientos similares a los de un sistema subamortiguado, en los que la curva de infectados se aproximaba a un valor constante final conforme transcurría el tiempo.

Al simular numéricamente el sistema antes propuesto se obtienen soluciones como la que se muestra en la figura 4.3.

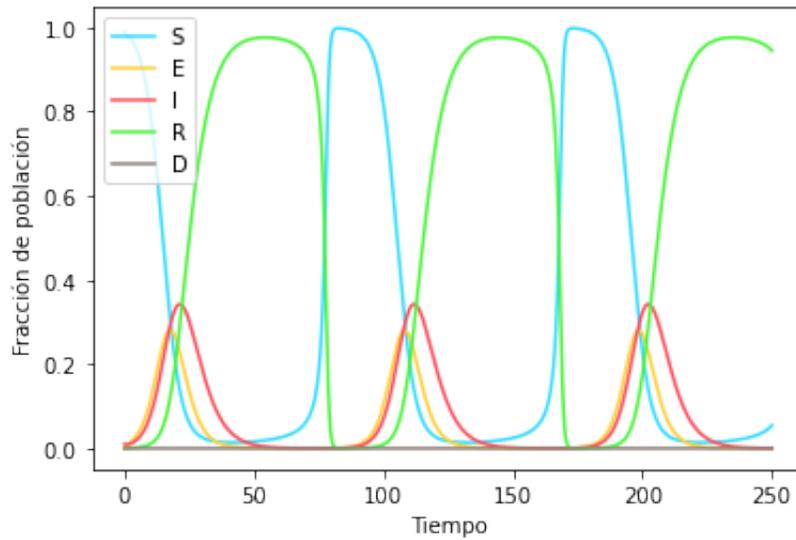


Figura 4.3: Ejemplo de soluciones cíclicas obtenidas al introducir el término no lineal propuesto para generar olas en las ecuaciones SEIRD. Los valores utilizados para los parámetros fueron $\alpha = 0.94$, $\beta = 0.3$, $\gamma_1 = 0.2$, $\gamma_2 = 0$, $\mu = 5$, $C_{\text{sat}} = 2$ y las condiciones iniciales $S_0 = 0.99$, $E_0 = 0$, $I_0 = 0.01$, $R_0 = 0$, $D_0 = 0$.

Puede apreciarse, que tal y como se buscaba, la modificación realizada resulta en la

introducción de patrones cíclicos en las soluciones. El término no lineal, expresado en 4.8, se adaptaría posteriormente para simular el mismo efecto en el modelo basado en agentes desarrollado, como se explicará en la sección 5.1.3.

4.2. Modelado de epidemias bajo una perspectiva de sistemas complejos

El modelo de ecuaciones SEIRD presentado anteriormente, conlleva ciertas limitaciones importantes, sobre todo respecto a los fines que se buscan en este trabajo. Quizás la más importante de ellas es que simula únicamente la evolución de grupos colectivos de varias personas, como es la fracción de infectados $I(t)$, mientras que en nuestro modelo se desea generar también series de tiempo de variables asociadas a personas individuales, como serían sus coordenadas de posición. Otra limitante a nombrar, es que en las ecuaciones SEIRD se hace tácitamente la suposición de que las personas pertenecientes a cada grupo colectivo se comportan exactamente del mismo modo, sin embargo en el modelo a construir aquí nos gustaría considerar diferencias en comportamiento que pueden presentar distintos miembros de una población. Estas restricciones llevaron a que se optara por un enfoque un tanto distinto para simular la pandemia.

El enfoque al que se recurrió fue el de los sistemas complejos. En este, la estrategia de modelado consiste en definir tanto las partes individuales del sistema a estudiar, como las interacciones que podría haber entre ellas, para posteriormente simular y observar la evolución del sistema en su totalidad. La ideología detrás de este método se basa en que hay ciertos sistemas cuyo comportamiento colectivo no se puede describir con reglas sencillas, ya que al hacerlo podrían omitirse comportamientos emergentes importantes, que surgen de la propagación de no linealidades en el sistema conforme las partes del mismo interactúan. Una forma en que podemos resumir lo anterior es decir que la representación mínima de un sistema complejo es el propio sistema, y por tanto, lo mejor que se puede hacer para estudiarlo es intentar modelar en forma fiel sus componentes, y observar cómo cambia el sistema conforme estas interactúan o se comunican entre sí.

En el caso particular de una pandemia, lo que se buscaría en términos computacionales es usar estructuras de datos apropiadas, que permitan describir tanto el estado de los individuos de una población (por ejemplo, guardando su estado de salud, posición, etc.), como las interacciones que puede haber entre dichas personas. Una vez hecho esto se puede evolucionar el sistema, y observar cómo va cambiando el estado de los individuos conforme interactúan propagando la enfermedad. Las siguientes secciones ilustrarán algunas de las

herramientas más importantes que pueden aplicarse para tal propósito. En última instancia, la que se usará en este trabajo es la modelación basada en agentes (MBA), sin embargo, convendrá también revisar otros dos modelos que son en realidad casos particulares de MBA: los autómatas celulares y las redes. Esto porque son modelos que ilustran adecuadamente como plantear en una computadora el fenómeno de una pandemia, además de que sus limitantes justifican la elección de usar una estrategia más general como son los agentes en esta tesis.

4.2.1. Autómatas celulares

En los autómatas celulares la estructura de datos que se suele utilizar es una matriz de celdas, donde cada una de las entradas representa un individuo distinto de la población. Tales celdas almacenan un número, que indica el estado de salud en el que se encuentra en ese momento la respectiva persona. Los valores posibles del estado de salud se indican por medio de enteros y corresponden a los estados considerados dentro de la notación SEIRD, i.e. se tiene la relación: $S \rightarrow 0$, $E \rightarrow 1$, $I \rightarrow 2$, $R \rightarrow 3$ y $D \rightarrow 4$. En principio, cada persona permanece inmóvil en su celda y únicamente puede tener contacto con los 8 vecinos que le rodean. Por ende, la representación de autómatas supone limitantes de movimiento importantes en los individuos. Para actualizar el autómata, en cada paso de evolución se revisa el estado de salud de cada individuo (celda), y dicho estado se modifica con ciertas probabilidades. En caso de que la persona se encuentre en estado susceptible y alguno de sus vecinos sea un infectado, su estado de salud pasará a expuesto con probabilidad α . Por otra parte, las personas expuestas pasarán a infectadas con probabilidad β y las infectadas se recuperarán o fallecerán con probabilidades γ_1 y γ_2 respectivamente ¹.

Dentro de la simulación es posible contar en cada paso de evolución el número de celdas que se encuentran en cada uno de los estados de salud, lo cual permite obtener gráficos SEIRD como el de la figura 4.1. En este caso, para obtener fracciones de población, se pueden dividir los conteos obtenidos para cada estado entre el número total de celdas del autómata.

En la figura 4.4 se ha ilustrado el estado del autómata (fila superior) y el gráfico SEIRD correspondiente (fila inferior) para 3 instantes de tiempo distintos, donde las celdas del autómata fueron coloreadas de acuerdo a su estado de salud.

¹Estas probabilidades son análogas en interpretación a las tasas que se utilizaron en el modelo de ecuaciones diferenciales SEIRD. La escala, sin embargo, es un tanto distinta a la que se utilizaba en ellas, ya que aquí las transiciones entre estados se consideran a nivel individual y no a escala de grupos colectivos de personas.

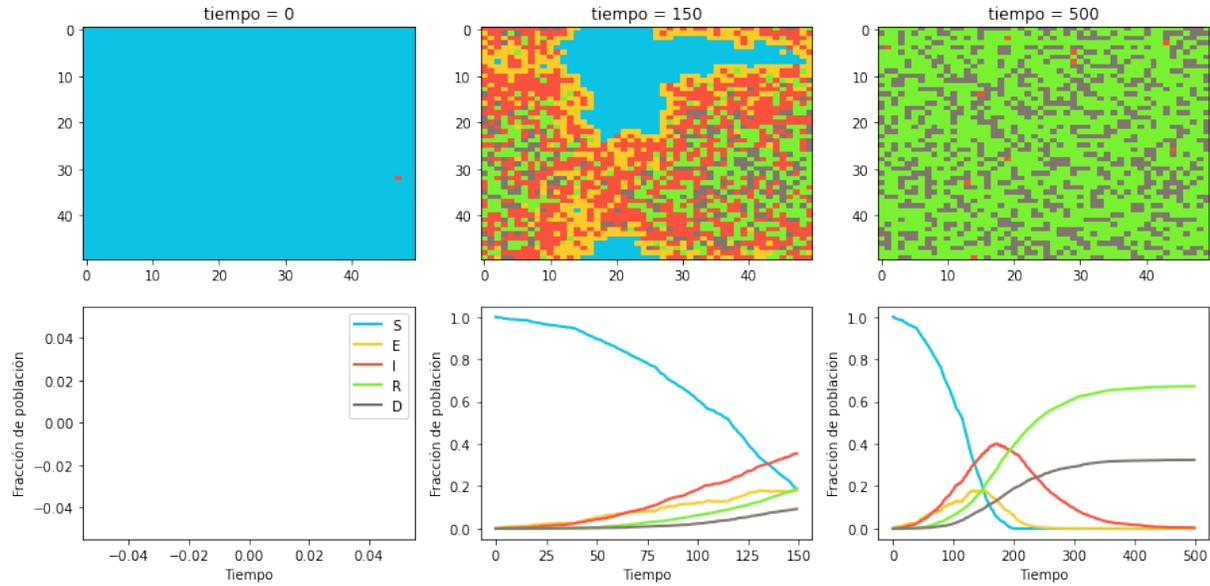


Figura 4.4: Propagación de una pandemia en un autómata celular de dimensión 100×100 en tres instantes de tiempo distintos $t = 0$, $t = 150$ y $t = 500$. Los valores utilizados para las probabilidades de transición fueron $\alpha = 0.9$, $\beta = 0.05$, $\gamma_1 = 0.01$ y $\gamma_2 = 0.005$.

4.2.2. Redes

Como se mencionó, en el caso de los autómatas, las personas con las que interactúa un individuo son siempre los 8 vecinos que le rodean. Esto representa una restricción considerable, ya que en un caso real los contactos que tiene cada persona pueden variar, y ello puede tener un efecto importante sobre la propagación de la enfermedad en la población. Considerar diferencias en los contactos de cada individuo puede además, ayudar a identificar las personas que sería conveniente aislar para frenar en forma efectiva la pandemia. Las redes o grafos constituyen una herramienta idónea para considerar tal variación en las conexiones de los individuos. En ellas, cada nodo podría representar una persona y los vértices trazados entre nodos denotarían que los dos individuos enlazados suelen tener contacto mutuo. Al igual que en el caso de los autómatas, cada nodo podría hallarse en uno de los estados $S \rightarrow 0$, $E \rightarrow 1$, $I \rightarrow 2$, $R \rightarrow 3$, $D \rightarrow 4$, y dicho estado iría cambiando a cada paso de la simulación con cierta probabilidad. Así, si un nodo susceptible llegara a encontrarse conectado con uno infectado pasaría a expuesto con probabilidad α , mientras que los expuestos transitarían con probabilidad β a infectados y los infectados a recuperados o fallecidos con probabilidades γ_1 y γ_2 respectivamente.

En la figura 4.5 se ilustran 3 instantes de la propagación de una enfermedad en una red de individuos, que en este caso se han conectado aleatoriamente para fines de demostración. La fila superior muestra el estado de la red, en la que nuevamente se ha utilizado un código

de color para identificar los distintos estados de salud y la fila inferior muestra la progresión del gráfico SEIRD hasta el respectivo punto de tiempo. En este caso, para obtener los datos de las series se realizó un conteo de los nodos que se hallaban en cada estado de salud en cada instante, y se dividió la cuenta entre el total de nodos de la red.

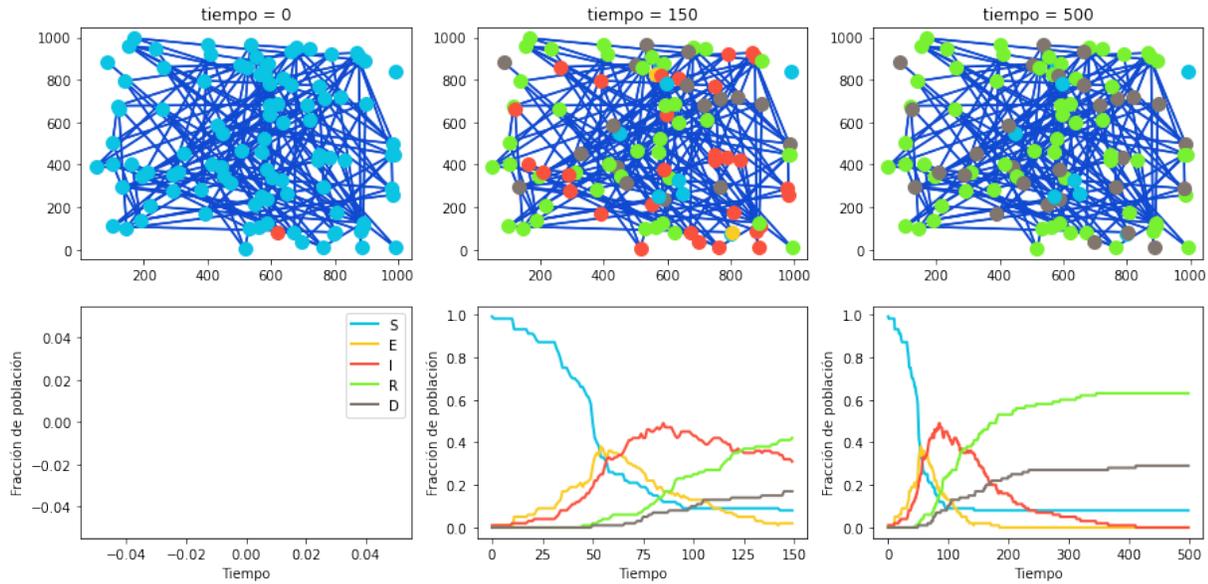


Figura 4.5: Evolución de una pandemia en una una red aleatoria de 100 individuos en tres instantes de tiempo distintos: $t = 0$, $t = 150$ y $t = 200$. Los valores utilizados para las probabilidades de transición fueron $\alpha = 0.9$, $\beta = 0.05$, $\gamma_1 = 0.01$, $\gamma_2 = 0.005$.

4.2.3. Modelos basados en agentes

El modelo de red explicado en la sección anterior ofrece una ventaja considerable, que es la posibilidad de variar las personas con las que tiene contacto cada individuo. Este enfoque, sin embargo, sigue presentando algunas limitantes a observar. Supongamos, por ejemplo, que se quisiera simular un caso donde los individuos fueran moviéndose por una ciudad. Adaptándonos a la representación de red, lo que se tendría que hacer es modificar las conexiones del grafo en cada paso de la simulación, conectando aquellos que se encuentren cerca espacialmente en ese momento. Lo anterior no solamente resultaría computacionalmente costoso, sino que incurriría en usos no deseables de las redes desde un punto de vista de estructuras de datos. Para este tipo de situación, donde se requiere que el vecindario de una persona cambie con el tiempo, resulta más adecuado utilizar una estrategia aun más general que las redes: la modelación basada en agentes (MBA). Esta es precisamente la que se utilizó en este trabajo, ya que se buscaba modelar la pandemia en

una ciudad, donde los individuos pudieran moverse a distintos lugares, e ir interactuando con varios grupos de personas conforme llevaban a cabo sus actividades cotidianas.

La teoría sobre agentes inteligentes fue introducida en una importante obra sobre inteligencia artificial escrita por Peter y Norvig [8]. En ella se define a un agente como una entidad que cuenta con sensores que utiliza para adquirir información de su entorno, y con efectores que usa para ejercer acciones que modifiquen su ambiente y/o su propio estado interno en respuesta al ambiente que percibe. El planteamiento de esta definición en la obra mencionada se hace en términos bastante generales, y realmente puede ser aplicada a distintos tipos de sistemas como son los humanos, robots, etc. Aquí, en particular, enfocaremos nuestra atención a agentes de software, que irán descritos por un conjunto de variables computacionales \mathcal{A} que definen su estado, sentirán un entorno digital mediante alguna función o rutina y podrán responder a su ambiente ejecutando otras funciones efectoras.

Concretamente en el caso de sistemas complejos y su simulación computacional, que es lo que nos interesa tratar en este trabajo, cada agente se caracterizará por un conjunto de variables internas \mathcal{A} y el entorno que percibirá será un conjunto de agentes vecinos \mathcal{N} con los que puede interactuar. Para plantear un modelo basado en agentes será necesario entonces definir dichos conjuntos, y posteriormente dar un conjunto de reglas de interacción, que especifiquen las acciones a realizar por parte del agente en función de su estado interno y el vecindario sentido. Si \mathcal{O} es un conjunto de acciones de salida posibles de los agentes, entonces se podría pensar a las reglas de interacción como mapeos que tienen la siguiente forma:

$$f : \mathcal{A} \times \mathcal{N} \mapsto \mathcal{A} \times \mathcal{O} \quad (4.14)$$

Algo a notar en la expresión previa es que se ha tomado en cuenta la posibilidad de que como resultado del mapeo se modifique el propio estado interno del agente. Además de esto, se establecerá en este momento que el conjunto vacío pertenezca a \mathcal{O} :

$$\phi \in \mathcal{O} \quad (4.15)$$

Esto permite que haya reglas de interacción "silenciosas", donde no se ejecute ninguna acción de respuesta como tal, y solo se modifique el estado interno del agente. De hecho, como se verá, la gran mayoría de reglas que se plantearán en los modelos de agentes serán de tal clase.

Los últimos párrafos de esta sección se dedicarán a ejemplificar cómo se plantea un modelo basado en agentes. Para mantener la discusión dentro de un contexto de epidemias,

se resolverá el problema de una pandemia, pero en una forma más simple a la que se ha desarrollado en la metodología de este trabajo.

El estado interno \mathcal{A} de cada agente contendrá algunas variables que caracterizan a cada individuo. Aquí, consideraremos como parte de dicho conjunto las coordenadas de posición del agente x, y , sus componentes de velocidad v_x, v_y , y su estado de salud, que será un número de 0 a 4, dependiendo de si el agente se halla en estado susceptible, expuesto, infectado, recuperado o fallecido. Al implementar dicho estado en una computadora puede resultar conveniente utilizar programación orientada a objetos, definiendo al agente como una clase que encapsule las propiedades del conjunto \mathcal{A} .

Por otra parte, el vecindario \mathcal{N} de un agente lo constituirán aquellos agentes que se encuentren a una distancia euclidiana menor que un radio R establecido. Obsérvese que como dicho vecindario puede cambiar conforme el individuo se mueve, es necesario calcularlo a cada paso de la simulación.

Una vez descritas las variables internas que definen a un agente y como se construye el vecindario que este es capaz de sensar, se pueden comenzar a enunciar las reglas de interacción del modelo. Para los propósitos de este caso ilustrativo serán suficientes las siguientes 2:

1. La primera regla describe como actualizar el movimiento de los agentes. En este modelo ejemplo las personas se moverán de manera aleatoria, por lo que se asignará en cada paso de evolución un vector de velocidad con dirección al azar, y con este se actualizarán las coordenadas de posición de los individuos. Para mantener a los agentes dentro de un área específica se usará la operación módulo en los cálculos de posición. Esto introducirá un efecto toroidal o "Pac-man", en el que al salirse del espacio rectangular donde habitan, los agentes reingresen por el extremo contrario. El algoritmo que describe esta regla se ha detallado en 4.1.

```

// A es el conjunto de variables internas del
// agente
// N es el conjunto de agentes vecinos del
// agente
Function movementInteraction(A, N):
    maxCompSpeed ← 10;
    // Establecemos velocidades en "x" y "y"
    // aleatorias que vayan entre 0 y 10
    A.vx ← random()*maxCompSpeed;
    A.vy ← random()*maxCompSpeed;
    // Actualizamos la posición del agente
    // worldWidth y worldHeight son
    // respectivamente el ancho y la altura del
    // espacio rectangular donde se mueven los
    // agentes.
    A.x ← (A.x + A.vx) %worldWidth;
    A.y ← (A.y + A.vy) %worldHeight;

```

Algoritmo 4.1: Regla de interacción para el movimiento aleatorio de un agente (Modelo ilustrativo de agentes).

2. La segunda regla actualiza el estado de salud de los agentes. Aquí la lógica sigue un camino muy parecido a la actualización de redes y autómatas que se explicó en secciones anteriores. Si el agente se encuentra en estado susceptible, y su conjunto de vecinos \mathcal{N} contiene algún infectado, este pasará con probabilidad α al estado expuesto. En forma similar, los agentes expuestos pasarán a infectados con probabilidad β y los infectados a recuperados o fallecidos con probabilidades γ_1 y γ_2 respectivamente. El algoritmo que detalla dicha regla es el que se muestra en 4.2.

```

// A es el conjunto de variables internas del
// agente
// N es el conjunto de agentes vecinos del
// agente
Function ruleHealthUpdate(A, N):
  if A.health == SUSCEPTIBLE and containsInfectedAgent(N)
  then
    // random() representa un número aleatorio
    // entre 0 y 1 muestreado de una
    // distribución uniformemente distribuida.
    if random() <  $\alpha$  then
      | A.health  $\leftarrow$  EXPOSED;
    end
  else if A.health == EXPOSED and random() <  $\beta$  then
    | A.health  $\leftarrow$  INFECTED;
  else if A.health == INFECTED then
    if random() <  $\gamma_1$  then
      | A.health  $\leftarrow$  RECOVERED;
    else if random() <  $\gamma_2$  then
      | A.health  $\leftarrow$  DEAD;
    end
  end

```

Algoritmo 4.2: Regla de interacción para actualizar el estado de salud del agente (Modelo ilustrativo de agentes).

En la figura 4.6 se muestran 3 instantes de la evolución del sistema de agentes explicado anteriormente. La fila superior despliega cada uno de los agentes como un cuadrado, cuyo color corresponde al estado de salud en el que se encuentra. Por otro lado, la fila inferior traza los gráficos SEIRD, contruidos contando a cada paso el número de agentes que se encontraban en los distintos estados de salud y dividiendo las cuentas entre el total de agentes.

Las dos reglas descritas en el modelo ejemplo definen cómo se mueven los agentes en su entorno y el modo en que se actualiza su estado de salud. En el modelo desarrollado en este trabajo existen también reglas que cumplen tales propósitos, pero que son un tanto más elaboradas en funcionamiento. Por ejemplo, respecto al movimiento, los agentes en

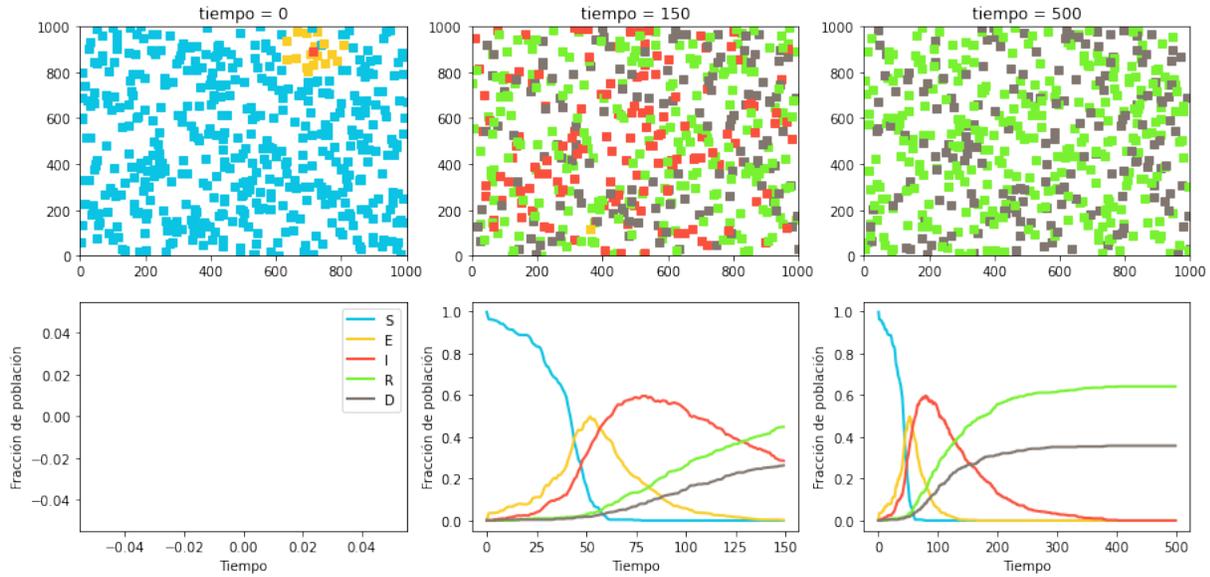


Figura 4.6: Evolución de una pandemia utilizando un modelo basado en agentes donde estos se mueven aleatoriamente en un espacio rectangular. El número de agentes utilizado fue 500, y los parámetros de infección fueron $\alpha = 0.9$, $\beta = 0.05$, $\gamma_1 = 0.01$ y $\gamma_2 = 0.005$.

la ciudad simulada no se mueven aleatoriamente, sino que van desplazándose entre los distintos sitios de la urbe conforme realizan sus actividades. Así mismo, la actualización del estado de salud considera ciertos elementos agregados, como son la separación entre infectados con síntomas y asintomáticos, la posibilidad de que al salir de un cuadro de enfermedad los agentes vuelvan a ser susceptibles y la pérdida de inmunidad de agentes recuperados debida a nuevas variantes del virus. Finalmente, se tienen ciertas reglas de interacción nuevas, como aquellas que indican la forma de actualizar el estado de actividad (rutina) de los agentes y las que establecen el modo en que afectan a los individuos políticas como el uso de mascarilla, aislamiento, etc.

Con ello finaliza nuestro tratamiento conceptual sobre el uso de agentes para simular sistemas complejos, dando paso ahora a una revisión de distintos modelos existentes consultados en la literatura.

4.3. Algunos trabajos y modelos existentes para simulación de pandemias

En esta sección se describirán algunos de los modelos existentes consultados, enfatizando sobre todo aquellos aspectos de los artículos que se trasladaron al modelo basado en agentes desarrollado.

4.3.1. COVID - ABS

COVID - ABS fue el modelo que motivó la decisión de implementar distintos escenarios de protección ante una pandemia en este trabajo, además de haber influido en la forma particular en la que se implementó uno de ellos: el distanciamiento social.

El propósito con el que se creó COVID - ABS fue estudiar como afectaban medidas como el aislamiento, el uso de cubrebocas y el distanciamiento social a la propagación de una pandemia y a la economía de una ciudad [9]. En dicho modelo se consideraban 5 tipos distintos de agentes que podían interactuar entre sí: personas, casas, empresas, gobierno y servicios de salud. En cuanto a las reglas de interacción de las personas podríamos mencionar que su posición se actualizaba usando una distribución normal, cuyo centro se ubicaba ya fuese en su hogar, lugar de trabajo o simplemente en su posición anterior. La elección del lugar alrededor del cual se movía un agente dependía sobre todo de la hora dentro del entorno virtual, simulando así una especie de rutina donde los agentes acudían a distintos lugares en ciertos horarios. Otra regla importante a mencionar es la actualización del estado de salud. Esta en realidad seguía un camino muy similar a la del algoritmo 4.2: se utilizaba una notación SEIRD para el estado de salud, y los agentes podían ir saltando de un estado a otro con cierta probabilidad. También en forma similar a dicho modelo ilustrativo, el contagio de agentes susceptibles dependía de que se encontraran a menor distancia que un radio R_I de otro agente infectado. En adición a las reglas de movimiento y actualización de estado de salud, COVID - ABS consideró una regla que simulara gastos monetarios de las personas. Así, si en algún momento el sujeto se hallaba suficientemente cerca de una empresa, se realizaba una transferencia de capital desde el agente hacia la misma, emulando una compra. Otras reglas a mencionar en las que se consideraron interacciones económicas entre agentes fueron el pago de salarios desde empresas a personas, el pago de impuestos de personas y empresas al gobierno y la financiación del gobierno al sector de salud.

Las políticas implementadas en COVID - ABS fueron 3: aislamiento, en el que se impedía el movimiento de todas o una fracción de las personas, uso de cubrebocas, en el que se disminuía la probabilidad de infección de los susceptibles, y el distanciamiento social, en el que el radio R_I de interacción entre personas decrecía. Dichas 3 políticas se tomaron como la base de escenarios a simular en este trabajo, aunque también a ellos se añadirían uno de vacunación y otro en la que se fusionaron varias de las medidas. La implementación que se realizó terminó siendo en realidad un tanto distinta a la de COVID - ABS en varios de los casos, a excepción de la política de distanciamiento social, en la que se imitó la idea de disminuir el radio de interacción R_I . Esto se explicará con mayor

detalle en la sección 5.1.6.

4.3.2. INFEKTA

INFEKTA es un modelo que permitió comparar la implementación de rutinas de personas propuesta en este trabajo con otra donde también se contemplaran, además de motivar la decisión de hacer una distinción entre casos asintomáticos y sintomáticos dentro de los estados posibles de salud de los agentes.

INFEKTA se creó con la intención de estudiar como la red de movilidad de una ciudad afectaba en la propagación de una enfermedad [10]. En dicho modelo, los agentes contemplados eran personas, a las que se les asignaba una rutina fija que realizarían todos los días dentro del ambiente simulado. Esta rutina consistía en trasladar al agente desde su hogar a un lugar de interés (escuela, trabajo, supermercado, etc.), y devolverlo después de un tiempo determinado a su casa. Uno de los elementos importantes de INFEKTA, es que la movilización de las personas se hacía dentro de una red de lugares, que conectaba las casas de los agentes, un sistema de transporte público (metro) y los lugares de interés de los individuos. Particularmente en los estudios presentados por los autores, dicha red se acopló a la distribución de lugares que hay en Bogotá, Colombia, incluyendo su sistema de metro llamado TransMilenio. En relación al estado de salud de los agentes, se siguió un sistema SEIRD, pero separando los agentes asintomáticos de aquellos que presentaban síntomas. Los contagios entre susceptibles e infectados eran posibles siempre y cuando se hubiesen encontrado en un mismo punto de la red de movilidad de la ciudad. Un detalle a destacar sobre la presentación de INFEKTA en el trabajo citado, es que los posibles cambios entre estados de salud los esquematizaron en forma muy conveniente con un diagrama de transiciones, algo que se decidió imitar en este trabajo mediante la Figura 5.5.

Los dos elementos principales que se tomaron de INFEKTA para esta obra fueron la separación entre sintomáticos y asintomáticos, y la forma visual en que decidieron representar la actualización del estado de salud en su artículo. Respecto a las rutinas de tareas de los agentes, podemos mencionar que la implementación realizada en esta tesis es un tanto más estructurada, ya que se consideran variaciones estocásticas en la secuencia de actividades de las personas y cambios que podrían darse con el tiempo en la conducta de los individuos. Así mismo, se añadió la posibilidad de calibrar la estadística de conductas de la población usando información de una encuesta (Véase 6.1.2). Las distinciones anteriores resultan comprensibles, ya que el enfoque de INFEKTA se centró sobre todo en simular en forma realista la red de movilidad de Bogotá, lo cual requirió considerar una gran cantidad de agentes y por lo tanto implementar en forma sencilla y eficiente las

rutinas de actividad de los habitantes.

4.3.3. Relación entre uso de mascarilla, transmisión asintomática y distanciamiento social en la propagación de COVID - 19

El artículo a tratar en esta sección motivó la forma en que se implementó la política de uso de mascarilla en este trabajo. La razón por la que se eligió esta referencia fue que se consideraban las diferencias en protección que puede brindar el cubrebocas, dependiendo de si era el agente susceptible o el infectado quien lo utilizaba, aspecto que no se contemplaba en otras investigaciones consultadas.

El modelo que se describe en esta sección se implementó con el propósito de examinar el efecto del uso de mascarilla, el distanciamiento social, y la fracción de asintomáticos en la pandemia de COVID - 19 [11]. Los agentes de este modelo se movían en forma aleatoria dentro de su entorno, imitando la regla de movimiento 4.1, mientras que para el estado de salud se utilizaba una clasificación SEIRD en la que se hacía una separación entre infectados sintomáticos y asintomáticos. Esto es similar a la forma en que manejaron el estado de salud los autores de INFEKTA, y enfatiza la importancia de considerar casos con pocos síntomas en una pandemia como la de COVID - 19, donde se tiene una amplia gama de niveles de reacción inmune ante el patógeno que se propaga. Al resultado de separar asintomáticos los autores le denominaron explícitamente como "SEIAR".

Para su implementación de la política de uso de cubrebocas los autores se basaron en un trabajo donde se investigó la reducción de transmisión viral cuando las personas usaban mascarilla [12]. En este, se concluyó que si era la persona infectada la que se protegía, se disminuía en un 95 % la transmisión, mientras que si era el susceptible quien portaba el cubrebocas la reducción en el contagio era del 85 %. Por otro lado, si ambas personas se protegían el efecto se combinaba, dando una eficiencia que para fines prácticos podía aproximarse a 100 %.

Los autores incorporaron las efectividades descritas en el párrafo anterior dentro de su modelo. Así, al inicio establecían cuales agentes llevarían cubrebocas a lo largo de la simulación, y cuando un susceptible y un infectado tenían contacto, la probabilidad de contagio α del agente propenso podía disminuir en 0 %, 85 %, 95 % o 100 %, dependiendo de cual(es) de los dos llevará puesta la mascarilla.

Esta forma de implementar la política de uso de cubrebocas se reprodujo en el modelo de agentes de esta tesis, con la diferencia de que en nuestras simulaciones los agentes no llevaban fijo el cubrebocas a lo largo de toda la simulación, sino que podían colocárselo o removerlo al entrar y salir de lugares. Para incorporar dicho dinamismo se utilizaron

distribuciones obtenidas con una encuesta, las cuales dictaban la probabilidad con que las personas utilizarían el cubrebocas en los distintos establecimientos de la ciudad (oficinas, supermercados, etc).

4.3.4. Distanciamiento social y resurgimiento de epidemias en modelos basados en agentes SIR

El modelo del que se hablará en esta sección influyó en la forma en que se implementó la política de aislamiento, en la que las personas reducen su movilidad como respuesta de resguardo ante una pandemia.

En el artículo a explicar se desarrollaban en realidad 3 modelos distintos para simular un aislamiento de personas [13]. Nosotros, sin embargo, nos centraremos en explicar únicamente su modelo de red, del que se tomaron las ideas esenciales para construir dicha política en este trabajo. Antes de comenzar con la descripción, sin embargo, es importante mencionar un conflicto de terminología que surgió al consultar otros trabajos en los que se implementaban escenarios de protección y resguardo. Muchos de ellos llamaban distanciamiento social a la situación de resguardo que aquí terminó denotándose como aislamiento. La razón por la que se optó en contra de esta convención es que se deseaba reservar "distanciamiento social" para referirnos a aquel escenario en donde las personas procuran mantener una distancia física mínima entre sí, situación que ciertamente es distinta a que los habitantes permanezcan en sus casas. Dicho lo anterior procedemos ahora a describir el modelo que nos concierne.

En el modelo desarrollado en el artículo los agentes iban moviéndose en los nodos de una red, teniendo la posibilidad de moverse con probabilidad d en cada paso a alguno de los nodos conectados con la posición en la que se encontraran. A diferencia de los grafos que se mostraron en la figura 4.5, donde los nodos se conectaban aleatoriamente, las redes de este modelo seguían una topología muy peculiar, en la que se tenían varios agrupamientos o comunidades de nodos suficientemente distanciadas entre sí. A este tipo de redes se les conoce como redes de mundo pequeño de Newman - Watts, y se tienen algoritmos bien definidos que permiten construirlas. En relación a la categorización del estado de salud, se utilizó una versión simplificada de SEIRD, en la que no se consideraban los expuestos y los fallecidos. A esta taxonomía se le conoce como SIR y en realidad se le podría considerar como un caso particular de SEIRD donde ciertas probabilidades de transición tienen valores iguales ya sea a cero o uno. Para actualizar el estado de salud se consideró que dos agentes entraban en contacto si se hallaban posicionados en nodos conectados de la red. Así, si uno era susceptible y el otro infectado había cierta probabilidad de que el susceptible se

contagiara pasando a infectado. Así mismo, en cada paso de la simulación las personas infectadas podían pasar con cierta probabilidad a recuperadas.

En este modelo la política de aislamiento se implementó reduciendo la probabilidad de que los agentes se movieran a nodos lejanos en la red de posiciones. Esta fue la idea que inspiró la implementación realizada en este trabajo para dicha política. Como se verá en la sección 5.1.6, en esta se simula el resguardo evitando que los agentes puedan hacer actividades en lugares que se encuentren más lejos que un radio establecido, lo cual es análogo a la restricción de traslado en conexiones muy largas que hacen los autores del artículo tratado aquí.

Con ello se termina con la introducción de conceptos de modelación basada en agentes y simulación de pandemias. Estas ideas constituyen la base con la que se planteó la metodología para construir nuestro modelo, el cual simula la movilidad de personas en una ciudad durante una pandemia. Las series de tiempo de la posición de agentes individuales se utilizaron a posteriori para examinar si era posible predecir o clasificar el estado de salud de una persona usando sus patrones de movimiento. Este análisis constituye la segunda parte de la metodología de este trabajo, y por tanto es momento de sentar las bases teóricas que lo sustentan.

4.4. Reconocimiento de actividad humana

El objetivo a tener presente en las secciones siguientes de este marco teórico, es el planteamiento de una metodología que permita obtener el estado de salud de una persona a partir de series de tiempo de sus coordenadas de posición $x(t), y(t)$. Esto, como veremos, requiere realizar dos transformaciones consecutivas. La primera de ellas llevará segmentos de las series de tiempo a un espacio de vectores de características, donde se facilita la identificación de puntos de tiempo en los que la persona se encontraba enferma. Hecho esto, la segunda transformación mapeará cada vector de características a una etiqueta, indicando si el individuo se encontraba sano (0) o enfermo (1) en el respectivo segmento de tiempo. Esta metodología se adhiere a los pasos de un área de la inteligencia artificial conocida como reconocimiento de actividad humana (HAR). El problema que se busca resolver en HAR puede enunciarse de la siguiente manera [14]:

Definición 1. *Dado un conjunto $S = S_1, S_2, \dots, S_n$ de series de tiempo de actividad de una persona, todas definidas en un lapso de tiempo $I = [t_\alpha, t_\beta]$, se busca encontrar una partición temporal I_0, \dots, I_r de I y un conjunto de etiquetas, que representen el estado o actividad del individuo durante cada intervalo de tiempo I_j . En esta definición los intervalos*

I_j son consecutivos, no vacíos, no se intersectan y su unión es igual al intervalo completo de tiempo, i.e. $\cup_{j=0}^r I_j = I$.

Dicho en otras palabras, lo que se buscaría es dividir las series de tiempo a analizar en ventanas consecutivas y asignar a cada una de ellas una etiqueta, que indique el estado o actividad de la persona en ese periodo de tiempo (Figura 4.7).

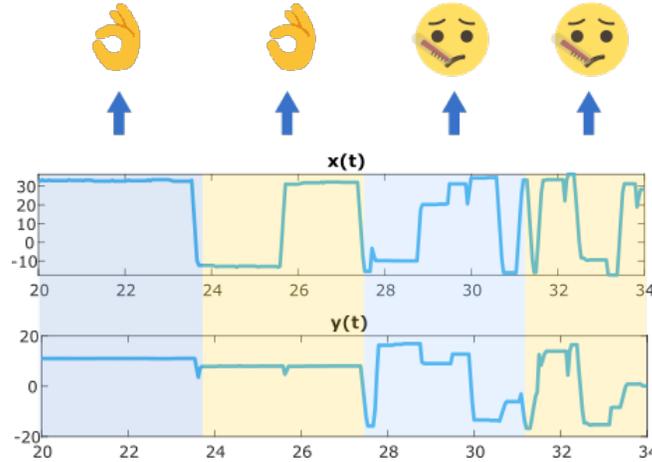


Figura 4.7: En HAR las series de tiempo de actividad de una persona se dividen en ventanas, y se asigna a cada una de ellas una etiqueta, que indica el estado o actividad de la persona en ese segmento de tiempo.

Concretamente en este trabajo las señales de actividad serán series de tiempo de las coordenadas de ubicación $x(t)$, $y(t)$ de personas, simuladas con el modelo basado en agentes implementado. Acorde a lo descrito anteriormente se buscaría dividir dichas series en ventanas y para cada segmento de tiempo decidir si la persona se encontraba sana (etiqueta 0) o infectada (etiqueta 1). Este ejercicio simularía una predicción del estado de salud de una persona usando una secuencia corta de sus datos de ubicación GPS.

Para resolver el problema planteado anteriormente se hacen dos transformaciones que describiremos a continuación. En la primera de ellas, se calcula para cada ventana un vector de características con los respectivos cortes de las señales de actividad. Para ejemplificar como podría hacerse esto considérese de nuevo la figura 4.7. El primer vector de características lo construiremos usando el recorte de las series $x(t)$, $y(t)$ que corresponde a la primera ventana, y que se ha delimitado con color azul en el diagrama. Lo que se busca es extraer información de dicho corte y almacenarla dentro de un vector. Así, podríamos establecer para un ejemplo sencillo que la primera entrada (característica) sea la media de $x(t)$ en la ventana y la segunda sea la media de $y(t)$, también en la ventana, con lo que tendríamos un vector de características de dos dimensiones para este primer segmento. Hecho esto, se calcularían las mismas características para los segmentos subsecuentes,

obteniendo en este caso 4 vectores de dimensión 2. Las cantidades a calcular se pueden elegir en principio libremente, sin embargo la intención es que en el espacio de los vectores de características sea más sencillo separar puntos que corresponden a los distintos estados que se buscan detectar (en nuestro caso entre sano y enfermo). En este trabajo se eligió utilizar características en los dominios de frecuencia y wavelet, ya que el movimiento de las personas suele seguir rutinas periódicas marcadas, que pueden verse afectadas si la persona se enferma.

La segunda transformación asigna a cada vector de características una etiqueta, que denota el estado o actividad de la persona en la respectiva ventana de tiempo. Dicho mapeo lo realiza un modelo de clasificación, cuyos parámetros internos deben primero entrenarse. El entrenamiento requiere utilizar un conjunto de vectores de características, cuyas etiquetas ya se conozcan de antemano. En nuestro caso, es posible obtener estas etiquetas de una serie de tiempo $H(t)$ del estado de salud de un agente. Como se verá en la sección 5.2, además de calcular el vector de características para cada ventana, se obtendrá su etiqueta "real" como la moda de $H(t)$ en el respectivo intervalo de tiempo, esto para posteriormente entrenar distintos clasificadores con el conjunto de vectores y etiquetas extraídas. Una vez entrenado el modelo de clasificación, este se puede utilizar para detectar el estado o actividad de una persona en series de tiempo nuevas, distintas a las que se ocuparon para entrenar. Los modelos de clasificación utilizados en reconocimiento de actividad humana son los característicos del aprendizaje de máquina, como son árboles aleatorios, máquinas de soporte vectorial, redes neuronales, entre otros.

Resumiendo las ideas de los párrafos anteriores, el reconocimiento de actividad humana involucra dos transformaciones: la primera lleva segmentos de series de tiempo a un espacio de características y la segunda lleva el espacio de características a un espacio de etiquetas por medio de un clasificador. Dicho flujo de trabajo se ha ilustrado mediante la figura 4.8.

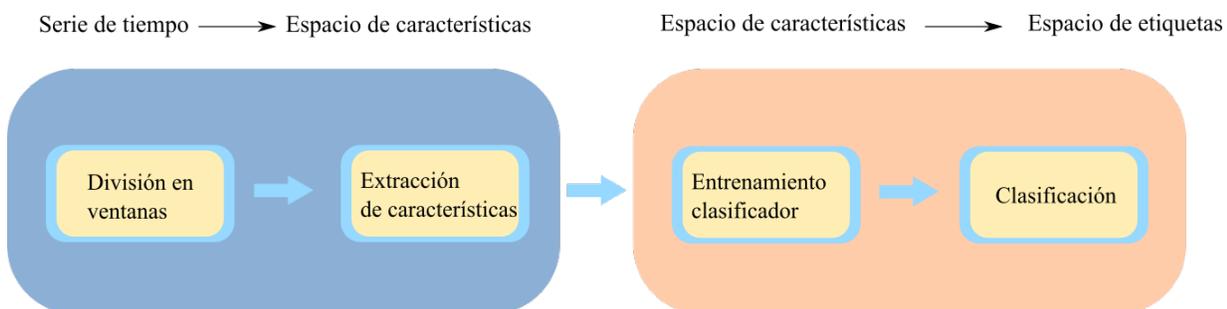


Figura 4.8: Flujo de trabajo del reconocimiento de actividad humana.

Las siguientes secciones se dedicarán a tratar los conceptos relevantes a estas transformaciones. Las características a extraer en nuestro caso son aquellas relativas al análisis de

señales en tiempo frecuencia, por lo que se hablará brevemente sobre las transformadas de Fourier y Wavelet. Así mismo se resumirá el funcionamiento de los distintos modelos de clasificación entrenados en este trabajo.

4.4.1. Características del dominio de la frecuencia y Wavelet

Como se mencionó anteriormente, las características que se obtendrán con los datos de movilidad simulados utilizan herramientas de análisis de señales en tiempo - frecuencia. Para justificar esta elección es necesario mencionar que existen estudios donde se ha observado que la movilidad de las personas presenta una alta regularidad y predictibilidad [15]. Lo anterior se debe en gran medida a la periodicidad que presentan las rutinas de actividad de las personas, y es por ello que extraer características en el dominio de la frecuencia puede resultar importante en la clasificación de este tipo de datos.

Las dos transformaciones que se utilizaron en este trabajo para obtener características fueron la transformada de Fourier discreta y la transformada Wavelet. Estas se explicarán en las siguientes subsecciones, enfatizando su aplicación a los datos de movilidad que nos conciernen.

Transformada de Fourier discreta

La transformada discreta de Fourier (DFT) transforma una señal $x[t_n]$ a una representación de la misma en el dominio de la frecuencia $X[\nu_k]$ [16]. Esta transformación se obtiene proyectando la señal en un conjunto de funciones armónicas (exponenciales complejas), por medio de la relación que se muestra en la ecuación 4.16.

$$\begin{aligned} \nu_k &= k(1/T) \\ X[\nu_k] &= \sum_{n=0}^{N-1} x[t_n] \exp(-2\pi i \nu_k t_n) \end{aligned} \quad (4.16)$$

Donde T representa la duración de la señal y ν_k representa la k -ésima frecuencia.

Un pseudocódigo sencillo para el computo de dicha transformación se muestra en 4.3.

```

Function DFT(x, t):
    // duración de la señal en tiempo
    T ← t[end] - t[0];
    N ← length(x);
    // Inicializamos el arreglo de frecuencias y la
    transformada
    freqs ← zeros(N, 1);
    transform ← zeros(N, 1);
    for k in [0,N] do
        freqK ← k*(1/T);
        value = ← 0;
        for n in [0,N] do
            value ← value + x[n]*exp(-2*pi*i*freqK*t[n]);
        end
        transform[k] ← value;
        freqs[k] ← freqK;
    end
    return freqs, transform;

```

Algoritmo 4.3: Cálculo de la transformada discreta de Fourier

La transformada de Fourier da información sobre las frecuencias presentes en una señal de estudio. Dicha información se suele representar en forma conveniente como un espectro de potencias, en el que se consideran únicamente la mitad de las frecuencias ν_k y se obtiene el cuadrado de la norma de la transformada $|X[\nu_k]|^2$ (Ecuación 4.17). Un aspecto importante a considerar es que como se recorta la transformada a la mitad, es necesario multiplicarla por 2 antes de obtener su cuadrado, para conservar la energía de la señal analizada. En el espectro de potencias los picos grandes corresponderán a frecuencias o periodicidades que son prominentes en la señal de estudio.

$$P[\nu_k] = |X[\nu_k]|^2 \quad (4.17)$$

Un punto pertinente a aclarar es la razón por la que únicamente se considera la mitad de las frecuencias que abarca el espectro. Cuando se aplica la transformada discreta de Fourier a una señal, se produce como resultado un arreglo en el que la mitad de los valores corresponden a la proyección con señales armónicas de frecuencia negativa. A pesar de que dichas frecuencias están bien definidas desde un punto de vista matemático, no

guardan como tal una interpretación física, y por ello los valores de la transformada que les corresponden suelen removerse al trabajar con datos con significado físico claro.

En este trabajo la característica que se obtuvo para cada ventana de las señales de movilidad fue la frecuencia promedio $\langle \nu_k \rangle$. Para obtener esta, se puede considerar al espectro de potencias $P[\nu_k]$ como una distribución de probabilidad si se le divide entre su suma. Una vez hecho esto puede hallarse la frecuencia promedio haciendo una suma ponderada de las frecuencias ν_k con su respectiva probabilidad (Ecuación 4.18).

$$\langle \nu_k \rangle = \sum_{k=0}^{N-1} \frac{P[\nu_k]}{\sum_n P[\nu_n]} \nu_k \quad (4.18)$$

En la figura 4.9 se muestran series de tiempo, construidas calculando la frecuencia promedio de segmentos consecutivos (ventanas) de señales de movilidad. Concretamente las señales analizadas en este caso fueron las componentes horizontal y vertical $x[t_n], y[t_n]$ de un agente que se movía por la ciudad virtual. En estos gráficos las secciones de tiempo donde el agente se encontraba infectado fueron indicadas en color rojo, mientras que los periodos donde la persona se hallaba sana fueron coloreados en azul. Para las pruebas, el tamaño de ventana utilizado fue en este caso de 50 puntos, lo cual, en tiempo de la simulación corresponde a usar datos GPS de la persona adquiridos en un periodo de 2.08 días.

Puede observarse que la frecuencia promedio de las señales de movimiento tiende a ser más alta y presentar menos variación cuando la persona se encuentra enferma, lo cual sugiere que dicha característica aporta información valiosa para la distinción entre los estados sano y enfermo.

Respecto a como se interpretan dichos gráficos, el valor de frecuencia promedio en el eje x representaría el número de oscilaciones promedio por día que tiende a presentar la coordenada x del agente al moverse por la ciudad, y lo correspondiente se tendría con la frecuencia promedio del eje y . Por lo tanto, un valor grande de dicha frecuencia indicará que el agente tiende a moverse frecuentemente durante el día. Bajo dicha interpretación es importante entonces, explicar porque los agentes enfermos tienden a moverse más que los sanos acorde a nuestros gráficos. La explicación yace en realidad en cómo se calibró la conducta de los agentes sanos. Como se verá en la sección 6.1.3, dicho comportamiento se especificó usando información de una encuesta de actividades, realizada en un periodo donde las personas hacían sus rutinas predominantemente en casa. Esto implicó que dentro del modelo los agentes sanos se quedarán bastante tiempo en su hogar, mientras que los enfermos se movieran más al acudir al hospital para recibir atención médica. Dicho aspecto en la movilidad es uno de los aspectos a mejorar sobre este trabajo, y se analiza con más

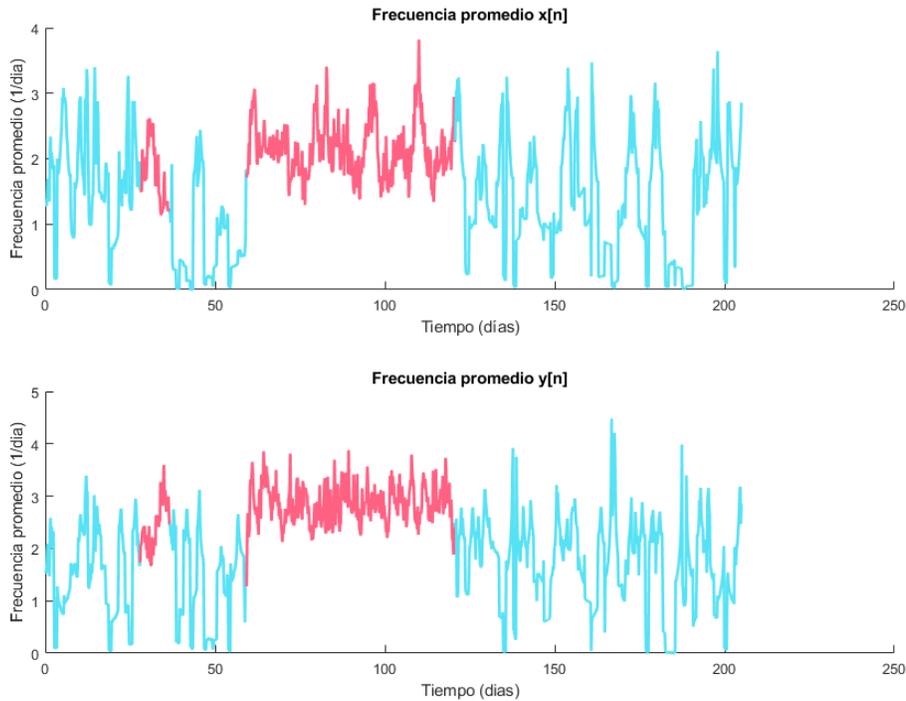


Figura 4.9: Serie de tiempo de la frecuencia promedio de las señales de posición de un agente. Las secciones en rojo corresponden a lapsos en que el individuo se encontraba enfermo. La unidad 1/día representa el número de veces por día que oscila la señal del agente en el respectivo eje.

detalle en la sección 7.1.2. Lo ideal sería calibrar el comportamiento de agentes sanos usando información recopilada en una situación de normalidad, donde las personas no se encontraran resguardados en casa ante el virus.

Transformada wavelet

La segunda transformación que se aplicó a las señales para extraer características fue la transformada wavelet. La formalización de esta herramienta involucra en realidad un tratamiento un tanto extenso, que no será posible replicar aquí. Por ello la discusión se limitará a explicar las ideas principales detrás de dicha transformación, y cómo es que se utilizó para obtener características de las series de tiempo de movilidad.

La transformada wavelet consiste en proyectar una señal $x(t)$ en un conjunto de señales ortogonales, construidas escalando y trasladando una misma función ψ que recibe el nombre de wavelet madre.

Si b es la traslación en unidades de tiempo de la wavelet madre y a mide el escalamiento de la misma, entonces la base de funciones en las que se proyecta $x(t)$ tiene la siguiente

forma:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right) \quad (4.19)$$

La idea es que se va proyectando la señal $x(t)$ en estas funciones, obteniendo un espectro $W(a, b)$, cuya magnitud nos dice a que escalas y traslaciones la señal analizada se parece más a la wavelet madre:

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi \left(\frac{t-b}{a} \right) dt \quad (4.20)$$

Establecer a la wavelet madre ψ como una función general hace que el análisis a realizar sea muy versátil. Como ejemplo, uno podría pensar en utilizar la forma de onda de una nota musical para detectar a qué escala y tiempo se presenta dicho tono en una señal de audio dada.

Es importante aclarar que la definición presentada en 4.20 para la transformada wavelet aplica estrictamente hablando solo para señales continuas $x(t)$, razón por la que se le da el nombre de transformada Wavelet continua (CWT). Existen formas de aplicar una aproximación de dicha operación a señales digitales almacenadas en una computadora, sin embargo aquí nos interesaría hacer una transformación que esté puramente definida para señales discretas. La razón por la que la CWT no es exacta en señales digitales es que no todos los muestreos de a y b producen wavelets ortogonales en el mundo discreto. Para solucionar esto lo que se suele hacer es definir wavelets que son ortogonales al escalarse a mitades. Con ello que se consigue una operación bien definida al usar escalamientos de forma $a = 2^{n/v}$, donde v es un entero fijo y traslaciones que sean múltiplos del intervalo Δt entre puntos sucesivos. Bajo tales condiciones la transformada Wavelet discreta (DWT)² se define de la siguiente manera:

$$W[n, m] = \frac{1}{\sqrt{2^{n/v}}} \sum_{k=0}^{N-1} x[t_k] \psi \left[\frac{t_k - m\Delta t}{2^{n/v}} \right] \Delta t \quad (4.21)$$

Una aplicación importante de la transformada wavelet discreta es la descomposición de una señal en diferentes escalas, lo cual se conoce como análisis multiresolución [17]. En este análisis lo que uno obtiene es una aproximación de la señal, que considera los cambios de la misma a una escala grande de tiempo, y algún número especificado de detalles, que reflejan variaciones de la señal que ocurren a escalas cada vez más pequeñas. En este trabajo, las características del dominio wavelet que se obtuvieron con las señales de movilidad fueron

²A partir de este punto toda vez que se haga mención de la transformada wavelet nos referiremos específicamente a esta definición discreta

las energías de la aproximación wavelet y de 3 detalles de descomposición.

Si $v = 1$ en la ecuación 4.21, entonces el detalle n -ésimo es de la siguiente forma:

$$D_n[t_k] = \frac{1}{\sqrt{2^n}} \sum_m W[n, m] \psi \left[\frac{t_k - m\Delta t}{2^n} \right] \quad (4.22)$$

Suponiendo que el número de detalles que se extrae es J , entonces la aproximación de la señal después de remover dichos detalles sería:

$$A_J[t_k] = \sum_{j=J}^{\infty} D_j[t_k] \quad (4.23)$$

El uso explícito de las ecuaciones anteriores para hacer la descomposición es generalmente computacionalmente costoso, además de que no puede llevarse a cabo en forma exacta, ya que 4.23 involucra una suma infinita. Resulta, sin embargo, que si las wavelet a utilizar se construyen en cierta forma ingeniosa, es posible obtener la aproximación y detalles por medio de la aplicación de filtros. No se hará aquí el desarrollo teórico a detalle, sin embargo, mencionaremos que lo que hay que establecer son una serie de M coeficientes a_i , que cumplen ciertas propiedades matemáticas [18]. Dichos coeficientes pueden pensarse como semillas o versiones comprimidas de la Wavelet madre y con ellos es posible reconstruirla en un proceso iterativo que va calculando cada vez más puntos de ψ . Esto sin embargo, no es necesario para hacer la descomposición en detalles y aproximación de una señal de estudio. Lo único que se requiere es un filtro pasa altas h y uno pasa bajas g definidos usando dichos coeficientes "semilla" a_i :

$$h = [a_{M-1}, a_{M-2}, \dots, a_0] \quad (4.24)$$

$$g = [-a_0, a_1, \dots, (-1)^M a_{M-1}] \quad (4.25)$$

La manera en que se aplican los arreglos h y g es a modo de filtros FIR. Por ejemplo para filtrar una señal x usando el filtro pasa altas h se haría:

$$x_{\text{filtrada}}[n] = \sum_{k=0}^{M-1} h[k] x[n-k] \quad (4.26)$$

La descomposición de detalles como tal, se realiza aplicando la cascada sucesiva de filtros que se ha esquematizado en la figura 4.10.

Supóngase que iniciamos con una señal $x[n]$. Es posible obtener de ella el primer detalle D_1 si se le filtra con el pasa altas h . Por otra parte, al aplicar el filtro pasa bajas g a la se-

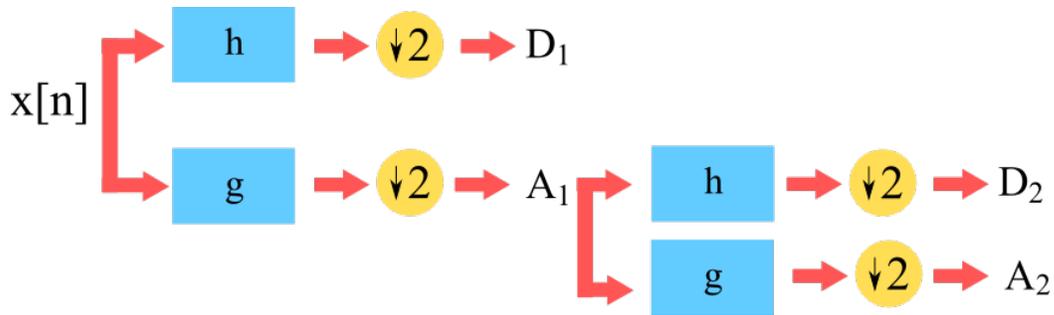


Figura 4.10: Cascada de filtros utilizada para descomponer una señal en su aproximación y sus señales de detalle en el análisis multiresolución con wavelets.

ñal, se extrae la primera aproximación A_1 . Cada vez que se aplica un filtro se submuestra a la mitad la señal para no tener información redundante. Lo anterior se puede hacer, por ejemplo, quedándonos únicamente con los datos en las entradas pares de la señal filtrada. Para descomponer el segundo detalle se aplica el filtro pasa altas a la primera aproximación A_1 y posteriormente se puede descomponer la segunda aproximación aplicándole a esta misma el filtro pasa bajas. El proceso antes descrito puede repetirse iterativamente con las nuevas aproximaciones hasta que se haya extraído el número de detalles deseado. Una implementación que se hizo de la transformada Wavelet discreta puede consultarse en [19]. Con ella se extrajeron la aproximación A_3 y 3 detalles de descomposición D_1 , D_2 , D_3 para el par de componentes de posición $x[t]$, $y[t]$. En dichos cálculos se omitió intencionalmente la reducción a la mitad en la cascada de filtros, porque se deseaba que las señales producidas por la descomposición mantuvieran el tamaño de la serie de tiempo original. La aproximación y los detalles fueron divididos en ventanas y se obtuvo la energía de cada uno de los segmentos. Dicha energía se calculó sencillamente como la suma de los cuadrados de la respectiva señal.

En la figura 4.11 se muestra una serie de tiempo de las energías calculadas para cada ventana de tiempo en las señales Wavelet descompuestas de la componente horizontal $x[t]$. Al igual que se hizo con la frecuencia media, los segmentos donde la persona se encontraba enferma fueron coloreados en rojo.

Puede observarse que las energías de los detalles se incrementan en la porción de tiempo donde el agente se encontraba infectado, lo cual reafirma que usar estas energías como características podría aportar información importante en la clasificación del estado de enfermedad de las personas.

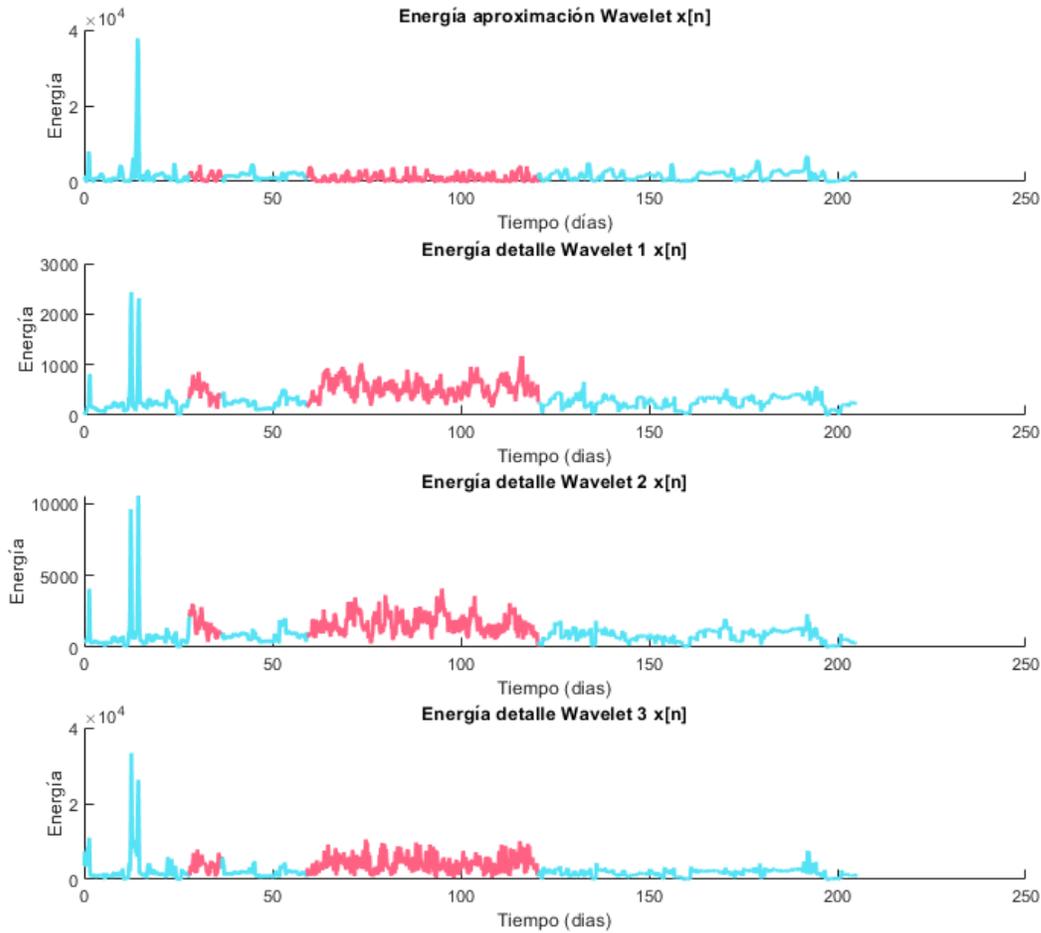


Figura 4.11: Series de tiempo de las energías de la aproximación Wavelet y 3 detalles de la señal de movimiento horizontal $x[t]$ de un agente. Los segmentos en color rojo corresponden a periodos donde el agente se encontró enfermo.

4.4.2. Modelos de clasificación

Ya explicadas las transformaciones que se aplicaron para llevar las series de tiempo a un espacio de características, se procederá a dar un breve resumen sobre los clasificadores utilizados para transformar los vectores de características en etiquetas, que indicarían si la persona se encontraba sana (etiqueta 0) o enferma (etiqueta 1).

Bosque aleatorio

El bosque aleatorio es un modelo de clasificación en el que se combina un número N_T de árboles de decisión distintos. Dado un vector de características \mathbf{x} , cada uno de los árboles

aporta una decisión respecto a la etiqueta a asignarle, y el bosque arroja como resultado final la etiqueta que haya recibido más votos, i.e. la moda del conjunto de decisiones de los árboles.

Cada uno de los árboles que conforman al bosque se entrena con una muestra de vectores de características escogidos en forma aleatoria (el muestreo se hace con reemplazo). Además de ello se toman de esos vectores un número reducido de entradas, es decir, se disminuye su dimensionalidad seleccionando un conjunto de características más pequeño que el inicial. Estas entradas, que se tomarán de los vectores para un árbol en particular son escogidas también aleatoriamente.

Así, si X es el conjunto completo de vectores de características que se tienen para entrenar y Y es el arreglo de las etiquetas asociadas a cada vector, al momento de entrenar el k -ésimo árbol, este recibirá un conjunto más pequeño de vectores con dimensión reducida X_k , junto con un arreglo de sus respectivas etiquetas Y_k . A continuación se dará una breve descripción de como se construye (entrena) cada uno de los árboles suponiendo 2 etiquetas 0 y 1:

1. Se obtiene la entropía:

$$S = P(Y_k == 1) \log \left(\frac{1}{P(Y_k == 1)} \right) + P(Y_k == 0) \log \left(\frac{1}{P(Y_k == 0)} \right) \quad (4.27)$$

Donde $P(Y_k == 0)$ es la probabilidad de que la etiqueta de los vectores sea cero y $P(Y_k == 1)$ es la probabilidad de que la etiqueta sea uno. Estas se obtienen contando el número de vectores a los que les corresponde la etiqueta en cuestión y dividiendo esta cuenta entre el total de vectores.

2. Suponiendo que la n -ésima característica también es binaria (incluso si no lo es, uno puede establecer por ejemplo que si su valor es mayor a cierto valor se asigne un uno y si es menor se asigne un cero), se dividen los vectores en 2 grupos: aquellos cuyo valor de la característica es 1, y aquellos cuyo valor para la característica es 0. Para cada una de estas 2 subtablas de vectores se obtiene la entropía de las etiquetas, como se hizo en la ecuación 4.27 pero en los conjuntos reducidos de vectores. Lo anterior dará un par de entropías, que denotaremos como $S_{n,1}$ y $S_{n,0}$. La ganancia de información IG_n para la n -ésima característica es:

$$IG_n = S - (S_{n,1} + S_{n,0})/2 \quad (4.28)$$

3. Se selecciona la característica M que dé la mayor ganancia de información, y esta se utiliza para hacer una bifurcación en el árbol.
4. Se repite el proceso con el subconjunto de vectores cuyo valor para la característica seleccionada era 1 (rama derecha) y con aquellos cuyo valor para la característica era 0 (rama izquierda). De esta forma se va incrementando el tamaño del árbol. Una rama dejará de bifurcarse cuando todos los vectores de un subgrupo tengan la misma etiqueta, o si se ha superado una altura máxima establecida para los árboles.

Una vez construido el árbol uno puede clasificar un vector dado \mathbf{x} recorriendo las ramas de acuerdo al valor de sus características hasta llegar a una de las hojas, que indicará la etiqueta que le corresponde.

Los parámetros a establecer en el entrenamiento de un bosque aleatorio son el número de árboles y la altura máxima que pueden tener estos.

Máquinas de soporte vectorial

Las máquinas de soporte vectorial son un modelo de clasificación que intenta optimizar la separación de vectores de características con etiquetas distintas. Lo que se busca expresamente es trazar una curva que separe en la mejor forma posible dichos puntos, de modo que se asigne una u otra etiqueta de acuerdo al lado de la curva en el que se halle el vector a clasificar. En principio la curva que se utiliza como frontera es un hiperplano (o recta si los vectores de características tienen 2 entradas), y se plantea como función a minimizar la distancia de dicho plano a los vectores que se encontrarán en lados opuestos del mismo (vectores de soporte). La optimización se realiza bajo una restricción importante, que es que la distancia de los puntos al plano de separación sea preferentemente mayor a una cantidad dada ϵ . En otras palabras, se intenta evitar que los vectores transgredan cierto margen alrededor del hiperplano.

El problema de optimización se plantea matemáticamente usando multiplicadores de Lagrange para combinar la función a optimizar y las restricciones. Una vez hecho esto, es posible construir el hiperplano solución resolviendo un sistema de ecuaciones, del que se obtienen los valores de los multiplicadores y los parámetros que definen a dicho hiperplano, ergo si este es de la forma $a_0 + a_1x_1 + \dots + a_nx_n = 0$, se obtendrán los coeficientes a_0, a_1, \dots, a_n .

En el problema anterior existe la posibilidad de que los puntos de clases distintas no se pueden separar adecuadamente con una plano en su espacio de características original. Por ello se cuenta con la opción de aplicar una transformación, que lleve a los vectores

a un nuevo espacio donde sí sea posible separarlos linealmente. A los distintos tipos de transformación disponibles que se tienen para aplicar se les da el nombre de Kernels. Tal nombre se atribuye a que en realidad no se aplica explícitamente la transformación, sino que se trabaja con una función $K(\mathbf{x}_1, \mathbf{x}_2)$ que para todo par de vectores en el espacio original devuelva su producto punto en el espacio transformado. A la sustitución de una transformación explícita de vectores por una función de este tipo se le suele nombrar dentro del aprendizaje de máquina como truco del Kernel (kernel trick).

Así, uno de los elementos importantes a establecer en el caso de las máquinas de soporte vectorial es el kernel a utilizar. Particularmente en este trabajo se utilizará el gaussiano, cuya ecuación es la siguiente:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma\|\mathbf{x}_2 - \mathbf{x}_1\|) \quad (4.29)$$

Donde γ es un parámetro a especificar.

Perceptrón multicapa

El perceptrón multicapa es una red neuronal que puede clasificar y aprender patrones de datos que no necesariamente son linealmente separables. Este consta de una capa de neuronas de entrada que procesa inicialmente los vectores de características a transformar, una o varias capas intermedias, y una capa de salida con una cantidad de neuronas igual al número de etiquetas que se puedan asignar. En nuestro caso dicha capa tendrá 2 neuronas únicamente, correspondientes a las etiquetas sano y enfermo. Así, cuando sea la primera etiqueta la que se asigne, la primera neurona se activará obteniendo como resultado $[1, 0]$ en la capa de salida, mientras que en el caso en que se asigna la segunda etiqueta el resultado será $[0, 1]$.

Computacionalmente la capa k -ésima se representa por una matriz \mathbf{A}_k y un vector de sesgo \mathbf{b}_k . El modo en que la capa procesa un vector de entrada $\mathbf{x}_{k,\text{in}}$ es el siguiente: en primer lugar la matriz se multiplica por el vector, y al resultado se le suma el vector de sesgo. Una vez hecho esto, se aplica a cada entrada del vector resultante de las operaciones anteriores un mapeo conocido como función de activación $\phi_k(x)$ (Ecuación 4.30). En principio cada capa puede ocupar una función de activación distinta, donde las comúnmente utilizadas suelen ser: sigmoide, lineal, ReLU, Softmax, entre otras.

$$\mathbf{x}_{k,\text{out}} = \phi_k(\mathbf{A}_k\mathbf{x}_{k,\text{in}} + \mathbf{b}_k) \quad (4.30)$$

El número de filas de la matriz \mathbf{A}_k corresponde a la cantidad de neuronas de la capa en cuestión y el número de columnas corresponde al número de entradas que procesan las

neuronas de la capa. En esta matriz a las entradas de una fila dada (neurona) se les puede interpretar como pesos, que indican que tanto responde la neurona en cuestión a cada una de las entradas que recibe. Esto es análogo al modo en que las neuronas del cerebro procesan los mensajes que reciben en sus distintas dendritas (entradas). Dependiendo de que tan fuerte sea la conexión en la dendrita en cuestión la neurona podría aumentar en mayor o menor medida su potencial como respuesta a la señal que reciba en ella. La función de activación es análoga al umbral de disparo que existe en las neuronas del sistema nervioso. Como se ha dicho, estas incrementan su potencial al recibir mensajes en sus dendritas, sin embargo este potencial puede disparar la neurona únicamente cuando se supera cierto umbral de activación. Así, en nuestra red de neuronas computacionales, multiplicar el vector de entradas por la matriz resulta en un vector que nos diría el incremento de potencial de cada neurona, el vector de sesgo establece un potencial base para cada neurona, por lo que para obtener el potencial total de estas hay que sumarlo al incremento producido por las entradas. Finalmente la función de activación corresponde a decidir si se dispara cada neurona de la capa.

Lo que se busca entrenar en una red neuronal son los pesos y sesgos de las neuronas, es decir, las entradas de \mathbf{A}_k y \mathbf{b}_k . Para ello, se calcula con cada vector de entrenamiento el error de la predicción de la red respecto a la etiqueta real, y este se utiliza para ir cambiando los pesos. Este proceso, conocido como backpropagation, es un tanto extenso, por lo que no se describirá a detalle, sin embargo mencionaremos que involucra obtener las derivadas de los pesos respecto al error, utilizando la regla de la cadena para modificar las capas más internas.

Dentro de una red neuronal los parámetros relevantes a establecer son el número de neuronas en las capas intermedias y las funciones de activación de cada capa.

Ensamble

Además de probar los modelos antes descritos para clasificar los datos de movilidad, se probó un ensamble que combinara varios de ellos.

Un ensamble es un conjunto de clasificadores entrenados que podríamos denotar como C . Cuando se le muestra a este un nuevo vector de características \mathbf{x} , cada uno de los clasificadores pertenecientes a C tomará una decisión respecto a la etiqueta a asignarle, y se arrojará como etiqueta final aquella que se haya repetido más veces dentro de las decisiones tomadas por los clasificadores. Dicha idea es en realidad la misma que se tenía en los bosques aleatorios, los cuales clasificaban los vectores usando votos de los distintos árboles de clasificación que los conformaban.

Con ello se culmina el marco teórico de este trabajo. Se han introducido hasta este punto algunos conceptos relativos a la modelación basada en agentes de epidemias, los cuales se utilizaron en el desarrollo de la simulación de movimiento de personas en una ciudad que se implementó. Los detalles de la metodología que se siguió para construir dicho modelo serán presentados en el siguiente capítulo.

Así mismo, se han introducido las herramientas requeridas para extraer características de los datos de movilidad simulados y clasificarlos. Estas se utilizaron para estudiar si era posible predecir el estado de salud de un individuo a partir de sus patrones de movimiento. La metodología relativa a dicha clasificación será también tratada en la siguiente sección.

Metodología

La metodología desarrollada en esta tesis fue dividida en 2 partes. En la primera de ellas se detalla un modelo basado en agentes, implementado para simular el movimiento de personas en una ciudad donde se propaga una enfermedad. Los datos de los habitantes generados con este modelo fueron utilizados posteriormente en una segunda parte de este trabajo, donde se estudió la predicción del estado de salud de una persona a partir de series de tiempo de su posición. En la figura 5.1 se muestra un mapa general de los temas a tratar en esta sección.

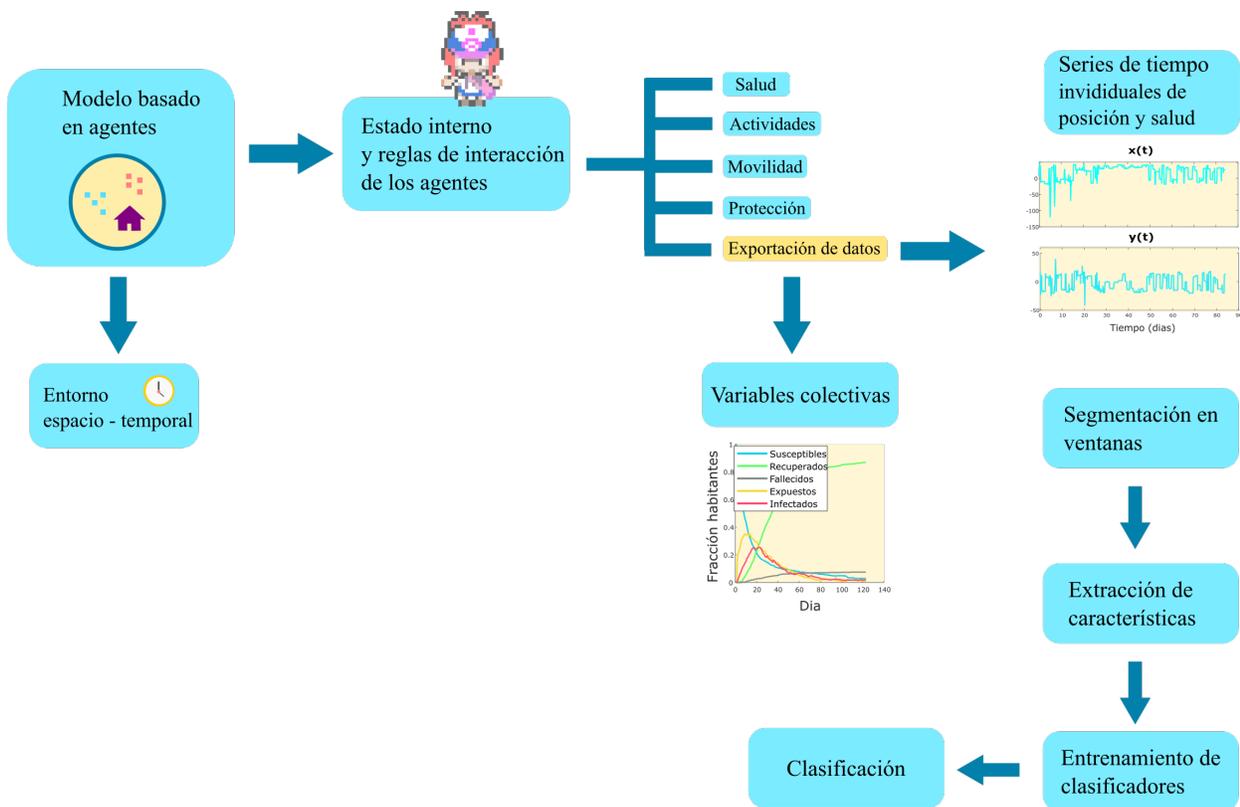


Figura 5.1: Esquema general de la metodología desarrollada

En lo que al modelo basado en agentes se refiere, se comenzará hablando sobre el software y lenguaje de programación utilizados para su implementación, y una vez hecho esto se explicarán los distintos elementos que componen al modelo. Sobre el modelo se describirá en primer lugar como se construyó la ciudad virtual en la que se desarrolla la simulación y cómo es que se maneja el paso de los días dentro de la misma, es decir, se establecerá el entorno espacio - temporal en el que evolucionarán los agentes. Posteriormente, se pasará a definir como tal los agentes que habitarán en dicho entorno, usando las ideas de modelado que se ejemplificaron en la sección 4.2.3. Recordando la definición de agentes provista, será necesario establecer el conjunto de variables internas \mathcal{A} que definen a cada persona, y un conjunto de reglas de interacción, que especifican como los agentes modifican su estado e interactúan entre sí con el paso del tiempo. En este trabajo las variables internas y reglas de interacción fueron separadas por conveniencia en 4 módulos distintos, que distinguen varios aspectos del comportamiento de un agente: salud, actividades, movilidad y protección. Sumado a los anteriores, se tiene dentro del programa un módulo global de exportación de datos, que permite recopilar y guardar en tablas series de tiempo de distintas variables del modelo. Las variables de interés a exportar con la simulación son de dos tipos. Las primeras, que llamaremos colectivas o macroscópicas, son cantidades que surgen de un comportamiento colectivo de varios agentes. Ejemplos de ellas podrían ser las series de tiempo SEIRD o la evolución de la fracción de personas que asiste a distintos lugares de la ciudad. Dichas variables serán de utilidad para comparar globalmente los distintos escenarios de protección simulados en el modelo (mascarilla, aislamiento, distanciamiento social, entre otros.). Dicha comparación tendrá el objetivo de determinar cual política consigue un mejor control de la propagación de la pandemia. El segundo tipo de variable que nos interesará obtener lo conforman aquellas que describen el estado de personas individuales, como son sus coordenadas de posición y su estado de salud. Estas, que podríamos denominar como microscópicas, constituirán los datos a usar para estudiar si es posible predecir el estado de salud de una persona a partir de su movimiento. La metodología utilizada para dicha tarea siguió los pasos de reconocimiento de actividad humana que se explicaron en la sección 4.4. En primer lugar, las series de tiempo de posición se segmentaron en ventanas, y para cada segmento se extrajo un vector de características. Las características utilizadas fueron en este caso cantidades que se obtienen de un análisis en tiempo - frecuencia de las señales, y por lo tanto fue necesario utilizar las transformadas de Fourier y Wavelet que se introdujeron en la sección 4.4.1. Una vez extraídos los vectores de características de cada ventana, se entrenaron y evaluaron con ellos los modelos de clasificación explicados en la sección 4.4.2. El objetivo que se persiguió fue que en función de el vector de características, el clasificador asignara una etiqueta, indicando si el agente se encontraba sano (0) o

enfermo (1) en la respectiva ventana de tiempo.

El problema de clasificación antes descrito se estudió tanto con datos obtenidos en un escenario donde los agentes no se protegían de ningún modo ante la pandemia, como en un caso de aislamiento, donde se resguardaban cuando se sobrepasaba un umbral en la ocupación hospitalaria. Esto para examinar si imponer políticas que restringieran el movimiento de las personas afectaba al poder de predicción del estado de salud a partir de datos de movilidad.

5.1. Modelo basado en agentes para simular movilidad de personas durante una pandemia

La implementación del modelo basado en agentes desarrollado se llevó a cabo con el motor de videojuegos Unity, en su versión 2021.2.9. Esta herramienta utiliza C# como lenguaje de programación, y un enfoque modular para asignar funcionalidades a los objetos de un juego o simulación. Algunas de las razones por las que se determinó que Unity era un software adecuado para implementar nuestro modelo basado en agente fueron las siguientes:

- Unity cuenta con funciones predefinidas para obtener y actualizar los vectores de posición de objetos que forman parte de un juego o simulación, lo cual brindaría facilidades al momento de implementar el movimiento de los agentes dentro de la ciudad virtual.
- Dado que Unity está destinado al desarrollo de videojuegos, cuenta con herramientas gráficas ya establecidas, que serían de utilidad para visualizar el movimiento de los agentes dentro de la ciudad durante las simulaciones.
- El enfoque modular que se utiliza para programar en Unity facilitaría la separación y organización de las variables internas de los agentes y sus reglas de interacción (véase 5.1.2). Esto fue importante, sobre todo para poder mantener una limpieza y orden en el código del programa, conforme se iban añadiendo nuevas funcionalidades a este.

Las siguientes secciones describen el modelo basado en agentes que se desarrolló, especificando el entorno virtual donde ocurre la simulación y la definición de los agentes que la componen. Si se desea, el código completo del programa puede consultarse en [20].

5.1.1. Entorno virtual

Sobre el entorno virtual será importante establecer las características del espacio (ciudad virtual) en donde se desenvuelve la simulación y como es que se maneja el paso del tiempo en la misma.

Ciudad virtual

El espacio donde ocurre la simulación es un área rectangular, en el que se distribuyen distintos tipos de establecimientos como son casas, parques, restaurantes, gimnasios y centros comerciales. El sistema de mapas de Unity (TileMaps) ofrece versatilidad en la colocación de estos lugares, y uno podría en principio diseñar una ciudad cuya distribución de locales se asemeje a la de alguna urbe real en concreto. En este trabajo, sin embargo, se decidió trabajar con una ciudad virtual sencilla, de tal forma que fuese fácil realizar un análisis de movilidad dentro de la misma. Dicha ciudad es la que se ha representado en la figura 5.2.

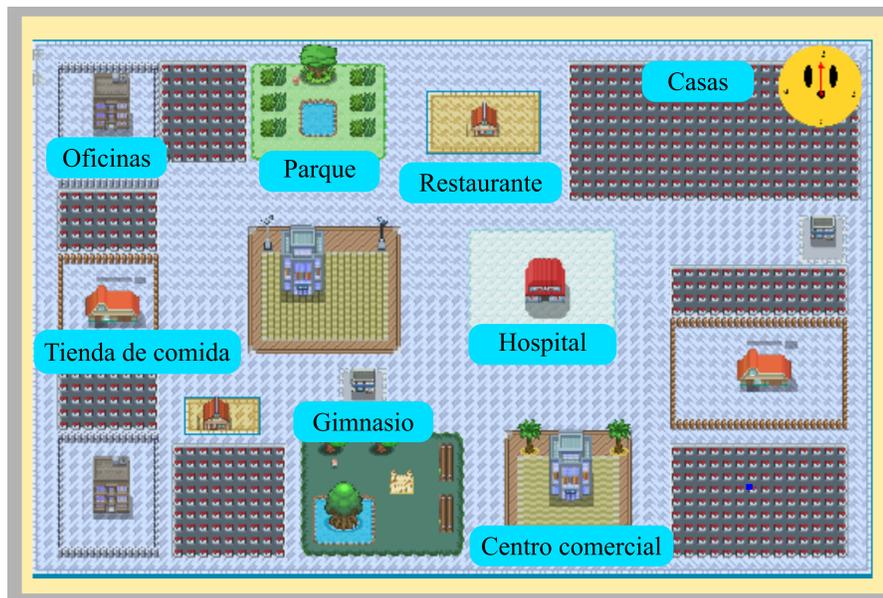


Figura 5.2: Mundo virtual donde se desarrolla la simulación

Entorno temporal

Para el manejo del tiempo dentro de la simulación se definió un parámetro δ_{clock} , que establece el tiempo real en segundos que corresponderán a una hora dentro del mundo simulado. El valor que se le dio a esta cantidad en las pruebas realizadas fue 0.7s/h,

lo cual significa que cada 0.7s habrá transcurrido una hora en la ciudad virtual. Este escalamiento de tiempo representa una cantidad fundamental en nuestra simulación, y como veremos afecta a la evolución de varias de las reglas de interacción que determinan el comportamiento de los agentes. Vale la pena, por tanto, enunciarle por medio de su propia ecuación:

$$\delta_{\text{clock}} = 0.7s/h \quad (5.1)$$

Dentro del modelo es necesario llevar un registro de la hora y día que acontecen en el mundo virtual, puesto que dichas variables llegan a ejercer un papel de regulación importante en los módulos de comportamiento de los agentes. Un ejemplo de dicho control se tiene en el caso de las actividades que realizan las personas, dentro de las cuales habrá algunas que no puedan hacerse en un horario nocturno, como lo es comprar comida. En 5.1 se muestra el algoritmo mediante el que se actualizan la hora y día en la simulación.

```
// Tcycle: Tiempo real en segundos que tomó el
// último ciclo del programa
// deltaClock: Factor de conversión de 0.7s por hora
// que pasa en la simulación
Function updateClock(Tcycle):
    // Vamos acumulando el tiempo real transcurrido
    // en la simulación
    realSeconds ← realSeconds + Tcycle;
    // Cuando se acumule el tiempo que corresponde a
    // una hora en la simulación (0.7s)
    if realSeconds > deltaClock then
        // Se reestablece el contador de segundos
        realSeconds ← 0;
        // Se actualiza la hora
        hour ← (hour + 1) % 24;
        // Se actualiza el día si ya es media noche
        if hour == 0 then
            | day ← day + 1;
        end
    end
```

Algoritmo 5.1: Algoritmo de actualización de día y hora dentro del mundo simulado.

5.1.2. Módulos de comportamiento de los agentes

Ya descrito el entorno virtual en el que se desarrolla la simulación, se comenzarán a definir los agentes que habitarán en el mismo. Recapitulando las bases teóricas sobre agentes que se explicaron en la sección 4.2.3, será necesario establecer 3 cosas: el conjunto de variables internas \mathcal{A} que define el estado de los agentes, el modo en que se construirá el conjunto \mathcal{N} de vecinos que un agente puede sentir, y finalmente un conjunto de reglas de interacción, que indicarán el modo en que se actualizará el estado del agente y las acciones que realizará en cada paso de la simulación.

Dado que el comportamiento de los agentes desarrollados en este trabajo es relativamente amplio, se decidió repartir sus variables internas y reglas de interacción en 5 módulos distintos, los cuales se han enumerado gráficamente en la figura 5.3.

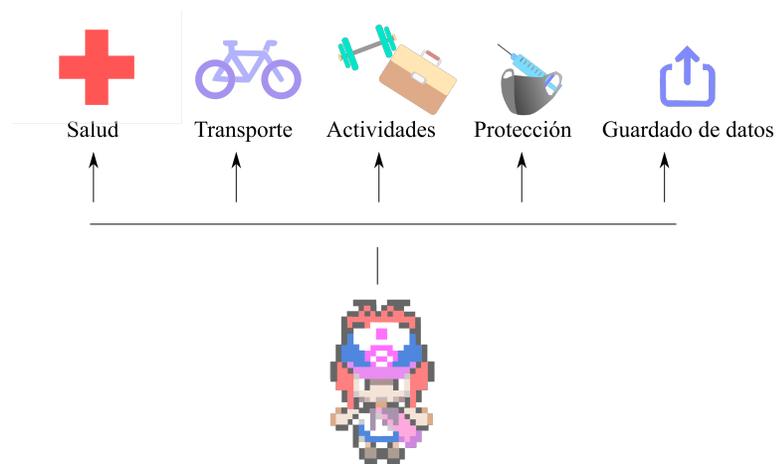


Figura 5.3: Módulos de comportamiento de los agentes

Las reglas de estos módulos son ejecutadas en cada ciclo de la simulación para todos los agentes. De ellos, la discusión aquí provista se enfocará exclusivamente en los primeros 4, ya que el módulo de guardado de datos no afecta directamente el comportamiento de los habitantes de la ciudad virtual. Este en realidad puede pensarse como una cámara o sensor, que sigue al agente en todo momento registrando con cierta frecuencia de muestreo algunas variables internas de este, como son sus coordenadas de posición y su estado de salud (Figura 5.4). Una vez que finaliza la simulación el módulo exporta un archivo de extensión `.csv` en el que se escriben en formato de tabla las series de tiempo recopiladas.

En la explicación provista para cada módulo se ha procurado seguir una estructura común, en la que se introduzcan primero las variables internas que abarca, para posteriormente establecer las reglas de interacción que le corresponden, ya sea por medio de pseudocódigos o diagramas de flujo.

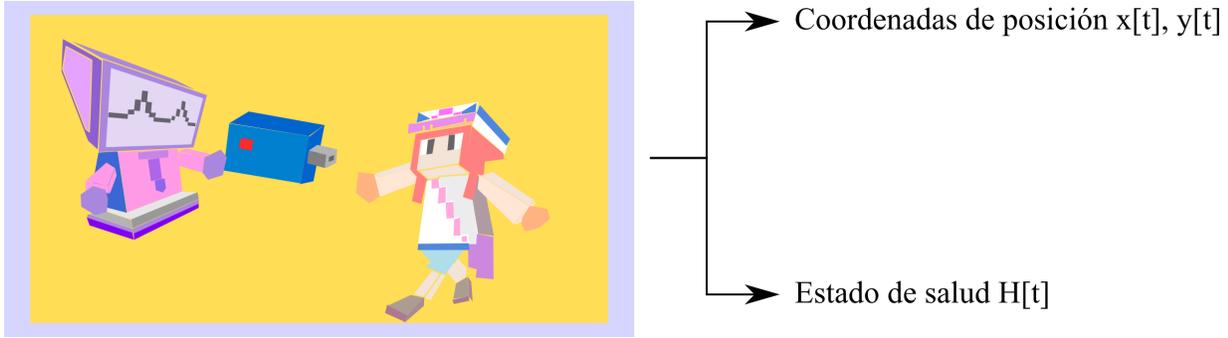


Figura 5.4: El módulo de guardado de datos puede pensarse como una cámara o sensor que sigue al agente en todo momento y recopila datos de su actividad, como son sus coordenadas de posición y su estado de salud.

5.1.3. Módulo de salud

Como su nombre sugiere, el módulo de salud es el que se encarga de guardar y actualizar el estado de salud de un agente. Los valores posibles para dicho estado corresponden a los estadios de enfermedad contemplados en los modelos SEIRD, sin embargo aquí se ha hecho una división del grupo de los infectados en dos tipos: aquellos que presentan síntomas nulos o leves, que llamaremos asintomáticos, y aquellos cuyo cuadro clínico se manifiesta en forma considerable, que nombraremos como sintomáticos. Bajo esta separación la lista de estados de salud posibles de un individuo sería la siguiente:

- Susceptible (S): Estado en el que el agente es propenso a contagiarse si llega a tener contacto con algún infectado.
- Expuesto (E): En esta fase el agente ha sido contagiado, sin embargo aun no es infeccioso, ya que el virus se halla en periodo de incubación.
- Infectado asintomático (I_A): La persona ya es contagiosa, pero presenta la enfermedad con síntomas leves o nulos, que no interfieren en forma significativa con sus actividades.
- Infectado sintomático (I_S): El agente es contagioso y presenta síntomas de gravedad considerable, que ameritan atención hospitalaria.
- Recuperado (R): La persona ha salido de un cuadro infeccioso y desarrollado inmunidad ante la variante del virus que lo atacó.
- Fallecido (D): El agente falleció después de haber pasado por un cuadro infeccioso.

La regla de interacción que actualiza el estado de salud del agente es similar en funcionamiento a la que se describió en la sección 4.2.3. En ella, los agentes pueden ir saltando a nuevos estados de salud con ciertas probabilidades. Un diagrama en el que se esquematizan las transiciones posibles entre los distintos estados de salud se muestra en la figura 5.5. Algo a aclarar es que las probabilidades de transición están definidas a nivel de agentes, y por tanto pueden verse modificadas para cada individuo por la acción de políticas u otras condiciones.

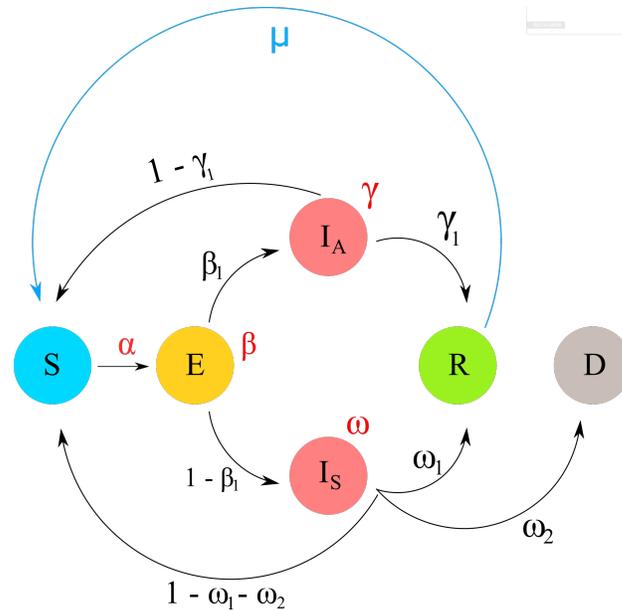


Figura 5.5: Transiciones del estado de salud de los agentes.

Las diferencias principales a mencionar sobre estas transiciones respecto a las que se manejaron en el sistema de ecuaciones 4.1 - 4.5, es que se añadió la posibilidad de que al salir de la infección los agentes volvieran a ser susceptibles de nuevo, además de agregar una posible vía desde los recuperados a los susceptibles. Esta última, resaltada en azul, corresponde a la aparición de nuevas variantes, y será explicada con más detalle más adelante en la sección 5.1.3 ¹.

En un inicio de la simulación se tendrá únicamente un agente en estado de infección asintomática, mientras que el resto serán susceptibles a contraer la enfermedad. Un agente susceptible podrá pasar a expuesto con probabilidad α toda vez que tenga contacto con

¹En análisis posteriores se examinó el efecto que tenía añadir tal retroalimentación desde el estado infectado al susceptible en las ecuaciones SEIRD. Lo que se halló fue que las soluciones mostradas en el marco teórico persistían, siempre y cuando la tasa de la transición agregada fuera relativamente pequeña respecto a las demás. Por ende, cumpliendo tal restricción nuestra comparación entre lo arrojado por el modelo de agentes y el modelo compartimental SEIRD presentado en el marco teórico continua teniendo validez.

alguna persona infectada, donde por contacto nos referimos a que el infectado pertenezca a su conjunto de vecinos \mathcal{N} . Una vez expuesto se podrá salir del periodo de incubación a cada paso de la simulación con probabilidad β , pasando ya sea a infectado asintomático con probabilidad β_1 o a infectado sintomático con probabilidad $1 - \beta_1$. Un infectado asintomático sale de su cuadro de enfermedad con probabilidad γ , y se tienen dos posibilidades. La primera es que el individuo genere inmunidad al virus, pasando con probabilidad γ_1 a recuperado. Por otro lado en caso de que no se generen defensas suficientes ante el patógeno se devolverá al agente con probabilidad $1 - \gamma_1$ a susceptible. Finalmente, los infectados sintomáticos podrán dejar de ser infecciosos con probabilidad ω , haciendo una transición hacia recuperados, fallecidos o susceptibles con probabilidades ω_1 , ω_2 y $1 - \omega_1 - \omega_2$ respectivamente.

Un aspecto importante que queda por detallarse es cómo se obtiene el conjunto de vecinos \mathcal{N} que puede sensar un agente, y que determina el contacto entre susceptibles e infectados. Para el cómputo de dicho conjunto se considerarán dos casos distintos. El primero es aquel en que el agente se encuentra trasladándose de un lugar a otro en las calles de la ciudad. En dicho escenario lo que se hará para obtener \mathcal{N} es sencillamente tomar aquellos agentes que también se encuentren moviéndose, y que se ubiquen a una distancia menor que un radio R_I del agente en cuestión. El segundo caso a considerar es aquel en que el agente se encuentra en un sitio de la ciudad, realizando alguna actividad. En tal situación será necesario definir el vecindario de forma un tanto más cuidadosa, ya que en la simulación no se ha programado explícitamente el interior de los lugares, y por tanto no es posible saber cuales agentes se encuentran cerca dentro de ellos. Lo que se hará es definir una variable F_p , que nombraremos factor de interacción, la cual indicará la fracción promedio de personas con las que interactúa un agente dentro del lugar p durante toda su estancia. Utilizando esta cantidad, la probabilidad en cada ciclo de la simulación de que un agente en el recinto se halle dentro del conjunto de vecinos del agente i -ésimo, también dentro, sería:

$$P_p(\text{Pertenencia a vecindario del agente } i\text{-ésimo}) = \left(\frac{F_p}{D_i}\right) \left(\frac{\Delta_{\text{cycle}}}{\delta_{\text{clock}}}\right) \quad (5.2)$$

Donde D_i es el tiempo en horas del mundo simulado que dura la actividad que está realizando el agente i -ésimo dentro del lugar, Δ_{cycle} denota el tiempo en segundos reales que tomó el último ciclo de actualización del programa, y δ_{clock} es el factor de conversión de tiempo enunciado en la ecuación 5.1. El factor $\left(\frac{\Delta_{\text{cycle}}}{\delta_{\text{clock}}}\right)$ puede interpretarse como el tiempo en horas dentro de la simulación que han pasado desde el último ciclo del programa, y al multiplicarlo se obtiene la probabilidad de pertenencia al vecindario por ciclo de la

simulación.

Los algoritmos antes descritos para la obtención de vecindarios y la actualización del estado de salud de los agentes han sido enunciados como pseudocódigos en 5.2 y 5.3.

```

// A es el conjunto de variables internas del agente
// deltaCycle es la duración en segundos del último ciclo del programa
// deltaClock es el factor de conversión entre tiempo real en segundos
// y horas dentro de la simulación
Function getNeighborhood(A):
  N ← [];
  // Si el agente se está moviendo en la ciudad
  if A.moving then
    for i in [0, nAgents] do
      if agents[i].moving and norm(agents[i].A.position - A.position) < RI then
        N.add(agents[i]);
      end
    end
  end
  // Si el agente se encuentra dentro de un lugar
  else
    // Probabilidad de pertenencia al vecindario (Ecuación 5.2)
    probNeighbor ← ((A.place.Fp)/(A.activity.duration))*(deltaCycle/deltaClock);
    for i in [0, nAgents] do
      if not agents[i].moving and agents[i].place == A.place and random()
      < probNeighbor then
        N.add(agents[i]);
      end
    end
  end
  return N;

```

Algoritmo 5.2: Algoritmo de obtención del vecindario de los agentes.

```

// A es el conjunto de variables internas del agente
// N es el conjunto de vecinos del agente
Function healthModuleRule(A, N):
  if A.healthState == SUSCEPTIBLE and containsInfected(N) then
    if random() <  $\alpha$  then
      | A.healthState  $\leftarrow$  EXPOSED;
    end
  else if A.healthState == EXPOSED then
    if random() <  $\beta$  then
      | if random() <  $\beta_1$  then
          | A.healthState  $\leftarrow$  INFECTED_ASYMPTOMATIC;
        else
          | A.healthState  $\leftarrow$  INFECTED_SYMPTOMATIC;
        end
      end
    end
  else if A.healthState == INFECTED_ASYMPTOMATIC then
    if random() <  $\gamma$  then
      | if random() <  $\gamma_1$  then
          | A.healthState  $\leftarrow$  RECOVERED;
        else
          | A.healthState  $\leftarrow$  SUSCEPTIBLE;
        end
      end
    end
  else if A.healthState == INFECTED_SYMPTOMATIC then
    r  $\leftarrow$  random();
    if r <  $\omega_1$  then
      | A.healthState  $\leftarrow$  RECOVERED;
    else if  $\omega_1 \leq r < \omega_1 + \omega_2$  then
      | A.healthState  $\leftarrow$  DEAD;
    else
      | A.healthState  $\leftarrow$  SUSCEPTIBLE;
    end
  else if A.healthState == RECOVERED then
    if random() <  $\mu$  then
      | A.healthState  $\leftarrow$  SUSCEPTIBLE;
      |  $\omega_2 \leftarrow \omega_2 * 0.5$ ;
    end
  end

```

Algoritmo 5.3: Regla de interacción para actualizar el estado de salud en la simulación.

Mutación y aparición de nuevas variantes

Uno de los elementos que se buscó simular en el modelo basado en agentes fue la aparición de nuevas variantes del virus que se propaga. En la naturaleza esto suele ocurrir por mutaciones que se dan en el material genético del virus, conforme estos utilizan la maquinaria celular de las personas que se contagian. Trasladado a nuestro modelo esto implicaría que los agentes perderán la inmunidad que adquirieron ante una versión previa del virus, i.e. pasarán de encontrarse recuperados a ser susceptibles a infectarse de nuevo. En el diagrama 5.5 esta transición se ha representado como un bucle resaltado en azul, y su probabilidad se ha denotado por la letra griega μ . Es importante mencionar que dicha probabilidad no permanece fija como el resto, porque establecerla de dicha forma no consigue el efecto periódico de oleaje en la curva de infectados que se deseaba incorporar. La forma que se le dio a dicha probabilidad resultó de adaptar el término no lineal que se introdujo en la sección 4.1.1 para el modelo SEIRD con ecuaciones diferenciales. Así, μ vendrá dada por la siguiente ecuación:

$$\mu = \mu_0 \left(\frac{S^2(t)}{C_{\text{sat}} + S(t)} \right) \left(\frac{\Delta_{\text{cycle}}}{\delta_{\text{clock}}} \right) \quad (5.3)$$

Donde $S(t)$ representa la fracción de agentes susceptibles, C_s es un parámetro llamado constante de saturación y μ_0 es otro parámetro, que determina la tasa de transición desde recuperados a susceptibles. Δ_{cycle} y δ_{clock} representan respectivamente el tiempo en segundos que tomó el último ciclo del programa y el factor de conversión presentado en la ecuación 5.1.

Cada vez que se hace una transición desde el estado recuperado al susceptible la probabilidad de fallecer ω_2 del individuo se reduce a la mitad, simulando una porción de la respuesta inmune que se irá transfiriendo entre las distintas variantes que surgen.

5.1.4. Módulo de actividades

El módulo de actividades guarda y gestiona las tareas que puede escoger el agente para realizar como parte de su rutina. Este módulo contempla dos reglas de interacción: una que indica como seleccionar y ejecutar nuevas actividades y otra que establece como irá modificándose con el tiempo la probabilidad de las distintas actividades disponibles.

Este módulo comprende una mayor cantidad de variables que el de salud, en el que solo era necesario mantener el estado de salud del agente. Por lo tanto empezaremos dando un listado de las variables de mayor relevancia que se utilizarán en las reglas de interacción a explicar.

- Lista de actividades posibles: El módulo guarda en primer lugar una lista, que contiene las tareas que el agente puede escoger para llevar a cabo en su rutina. A esta lista la nombraremos dentro de esta sección como `listActivities`. En concreto, las actividades posibles consideradas en este trabajo fueron comer, dormir, trabajar, comprar comida, socializar, ejercitarse e ir al médico. Cada actividad cuenta con varias propiedades que la distinguen, como son su duración, los tipos de lugar donde se realiza, el horario en que se puede llevar a cabo, su probabilidad e incrementos a utilizar para modificar en el tiempo dicha probabilidad.
- Bandera de ocupación: Para señalar que el agente se encuentra realizando una actividad se utiliza una variable booleana llamada `occupied`. Esta como veremos se usa en la regla de selección de actividades para indicar que el agente ha culminado su tarea previa y está listo para elegir una nueva.
- Actividad actual: La actividad que se encuentra desarrollando un agente en un momento determinado se guarda como un objeto llamado `activity`. Dicho objeto se utiliza para saber si se ha excedido la duración de la tarea que está realizando el agente, punto en el que este se desocupará. Además de ello, la duración de la actividad se emplea en la obtención de los vecindarios dentro de un lugar, como se indicó en la ecuación 5.2.
- Lugar de la actividad actual: Otro objeto que se almacena en este módulo es el lugar donde el agente se encuentra realizando una tarea, variable que nombraremos como `place`. Esta de hecho se introdujo ya tácitamente dentro del algoritmo de obtención de vecindarios presentado en 5.2. En el modelo cada lugar cuenta con distintas propiedades que lo caracterizan, como son sus coordenadas de posición, el tipo de lugar (supermercado, oficina, etc.), su factor de interacción F_p y el conjunto de agentes que se hallan dentro.
- Tiempo de ocupación: Finalmente, el tiempo que lleva un agente ocupado desarrollando una actividad se irá acumulando dentro de una variable flotante que llamaremos `activityTime`

Ya descritas las variables internas del módulo, comenzaremos a enunciar las reglas de interacción del mismo. La primera de ellas indica como el agente seleccionará y ejecutará nuevas actividades en su rutina. Esta se ha resumido con el diagrama de flujo que se muestra en la figura 5.6.

Cuando un agente se encuentra desocupado lo que hará es elegir como nueva tarea aquella actividad que tenga la mayor probabilidad en ese momento y que pueda realizarse

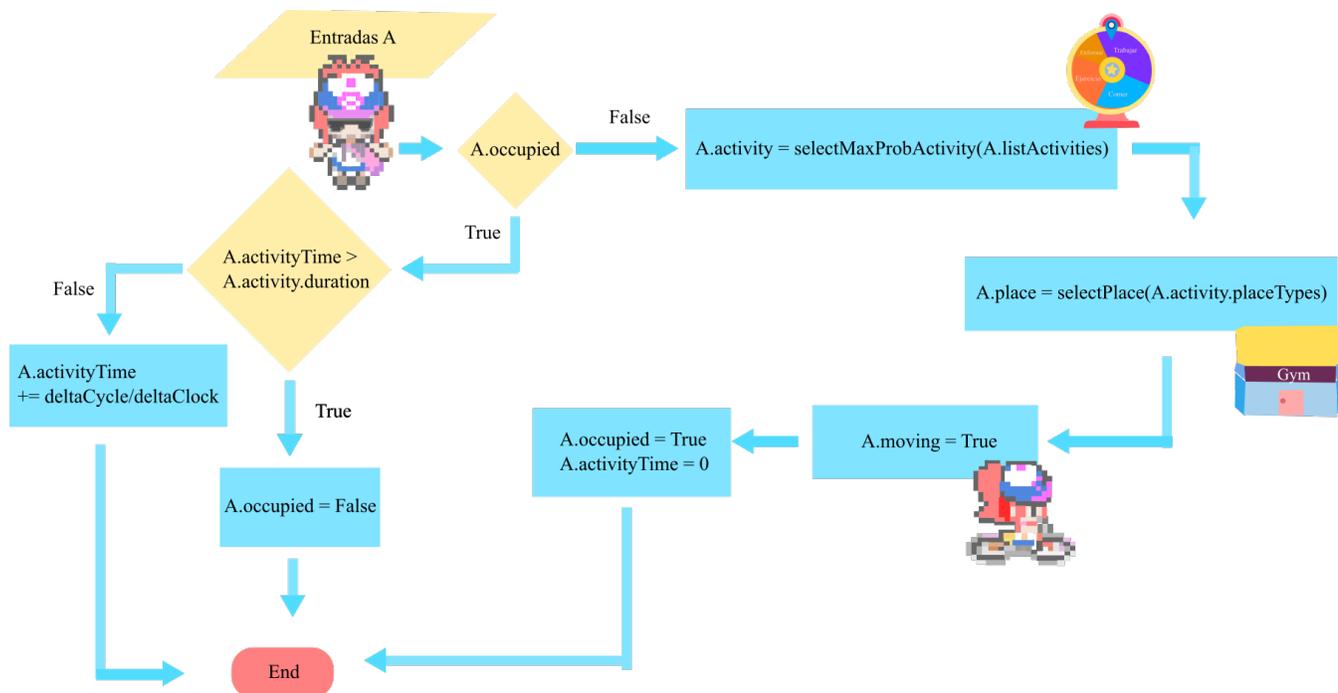


Figura 5.6: Regla de interacción para la selección y ejecución de actividades

en esa hora del día, guardándola dentro de la variable `activity`. Una vez hecho esto será necesario determinar el lugar de la ciudad donde se realizará la actividad que se eligió. Como se mencionó antes, cada actividad cuenta con una lista de tipos de lugar donde se pueden realizar, por lo que se obtendrá aleatoriamente un recinto de la ciudad cuyo tipo sea de entre los posibles, almacenándolo en la variable `place`. Es pertinente mencionar que la probabilidad de elección de los distintos tipos de lugar no es uniforme, sino que sigue una distribución obtenida con una encuesta (Véase 6.1.2). En dicho cuestionario se preguntó a los participantes en qué lugares solían hacer más a menudo las actividades consideradas, y las frecuencias relativas de las preguntas fueron usadas como probabilidades de selección para los tipos de lugar del modelo. Una vez seleccionado el lugar se indica al módulo de movilidad que hay que transportar al agente hacia este estableciendo como verdadera una variable `moving`. Finalmente, se señala que el agente ahora se encuentra ocupado volviendo el booleano `occupied` verdadero y se reinicia el acumulador de tiempo de ocupación a cero. Por otro lado, en caso de que el agente se halle ocupado se verificará si su estancia en el lugar ha excedido ya el tiempo de duración de su actividad. En caso de que sea así, se indicará que ahora el agente se encuentra desocupado y listo para elegir una nueva tarea. En caso contrario se acumulará el tiempo que ha pasado desde el último ciclo en la variable `activityTime` y se finalizará con la actualización.

En la regla de actualización antes descrita se mencionó que la actividad a seleccionar

como nueva era la que tuviese mayor probabilidad en ese momento. La segunda regla de interacción que explicaremos para este módulo define cómo evolucionará la probabilidad de las tareas disponibles a escoger.

Inicialmente todas las probabilidades se inicializarán en cero, y su probabilidad ira aumentando linealmente con una tasa dada. Si denotamos a la probabilidad de la i -ésima actividad como $P_{act,i}$, y a su tasa de incremento por hora dentro de la simulación como $\Delta_{act,i}$, entonces la actualización será de la siguiente forma:

$$P_{act,i}^{next} = \min\left\{P_{act,i}^{prev} + \Delta_{act,i} \left(\frac{\Delta_{cycle}}{\delta_{clock}}\right), 1\right\} \quad (5.4)$$

Nótese de dicha expresión que el incremento lineal de la probabilidad se hará hasta alcanzar el valor máximo de esta, que es uno, y posteriormente se mantendrá dicho valor. Cada vez que se realiza una actividad su probabilidad se devuelve a cero, ya que es improbable que el agente repita la misma tarea justo después de haberla culminado.

En la parte superior de la figura 5.7 se muestran series de tiempo de la probabilidad para dos actividades: trabajar y ejercitarse. Puede notarse que en efecto estas crecen en forma lineal, regresando a cero en aquellos puntos donde el agente realizó la respectiva actividad. Otro aspecto importante a observar en dicha figura es que la tasa de incremento de la probabilidad determina que tan a menudo se repetirá una tarea. En el caso ilustrado, la tasa de crecimiento de "trabajar" es claramente mayor que la de "ejercitarse", por lo que el agente ejecuta dicha acción más frecuentemente. Esta observación es importante, porque implica que los incrementos de probabilidad codifican una suerte de personalidad para el agente, determinando que actividades prefiere realizar en su rutina.

Dentro del modelo se consideraron dos incrementos de probabilidad posibles para cada actividad. El primero, que nombraremos como `deltaProbHealthy`, se aplicó para actualizar la probabilidad de la actividad en caso de que el agente se encontrara sano o asintomático, mientras que el segundo, llamado `deltaProbSick`, se aplicó cuando el agente se encontraba infectado con síntomas.

Para considerar diferencias en la conducta que puede haber entre los individuos sanos de una población, el incremento `deltaProbHealthy` de cada actividad se muestreó para cada agente usando información de la encuesta que se explicará en la sección 6.1.2. Poniendo como ejemplo la actividad comer, se preguntó a los encuestados cuantas veces al día solían alimentarse, dando como opciones [1, 2, 3, 4, 5, 6]. Estas corresponden a incrementos de probabilidad por hora que tienen valores [1/24h, 2/24h, 3/24h, 4/24h, 5/24h, 6/24h], así que se pueden utilizar las frecuencias relativas de la pregunta para asignar alguno de estos valores como el incremento de probabilidad de la actividad comer en un agente. Por

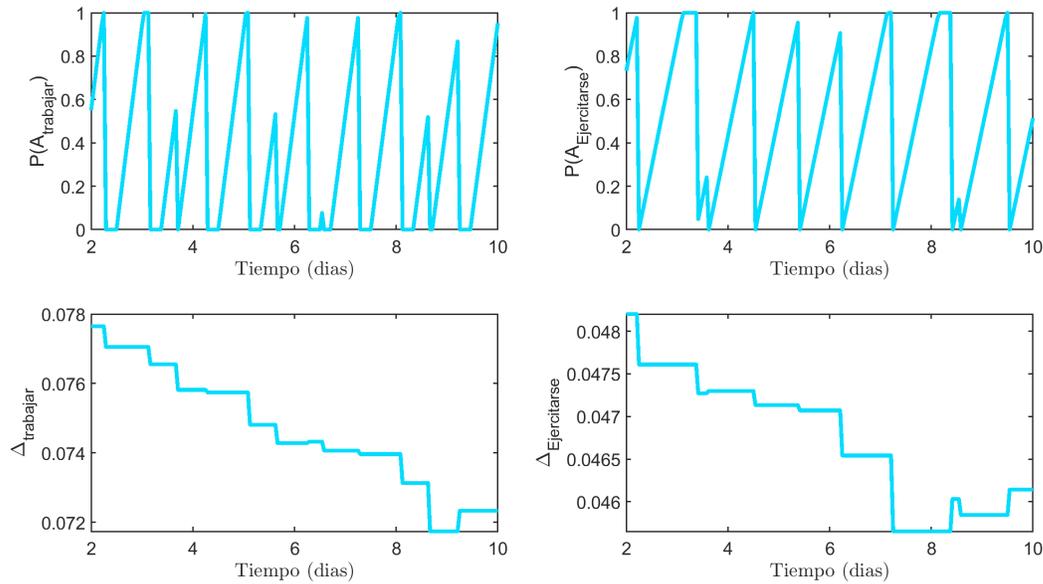


Figura 5.7: Variación temporal de la probabilidad e incremento de probabilidad para dos actividades distintas: trabajar y ejercitarse.

otra parte el valor del incremento `deltaProbSick` en cada actividad se asignó de igual forma para todos los agentes, ajustando empíricamente las cantidades en tal forma que el agente solo llevase a cabo con frecuencia tareas que suelen asociarse a un estado de enfermedad, como son ir al hospital, comer y dormir.

La parte inferior de la figura 5.7 muestra una serie de tiempo de los incrementos de probabilidad de las actividades trabajar y ejercitarse. Puede notarse que cada vez que se realiza la tarea en cuestión se modifica un poco dicho incremento. Esto se hizo para simular cambios pequeños que pueden ocurrir con el tiempo en la rutina de un individuo, haciendo que realice ciertas actividades más a menudo y otras con menor regularidad. La modificación de los incrementos se hace utilizando ruido Perlin, un tipo de ruido que se caracteriza por presentar persistencia a patrones de subida y bajada en los valores aleatorios que produce. Dicha herramienta fue desarrollada por Ken Perlin, quien buscaba desarrollar un método para generar superficies realistas mientras trabajaba en los efectos especiales de la película *Tron*. La técnica ha sido ampliamente adoptada, y resulta idónea para simular el tipo de variaciones aleatorias que suelen presentarse en fenómenos naturales [21].

El pseudocódigo de la regla completa de interacción para actualizar la probabilidad de las actividades se muestra en 5.4.

```

// A es el conjunto de variables internas del agente
// N es el vecindario del agente
Function ruleProbabilityUpdate(A, N):
    // Cuando el agente termina de realizar una actividad se reinicia
    // la probabilidad de esta a 0. Además de ello se modifica un poco
    // su incremento de probabilidad con ruido Perlin
    if not A.occupied then
        A.activity.probability ← 0;
        if A.healthState == INFECTED_SYMPTOMATIC then
            addPerlinNoise(A.activity.deltaProbSick);
        end
        else
            addPerlinNoise(A.activity.deltaProbHealthy);
        end
    end
    // Actualizamos la probabilidad de las actividades que sean
    // distintas a la que se está realizando/acaba de realizar
    // (Ecuación 5.4)
    for activity in A.listActivities do
        if activity ≠ A.activity then
            if A.healthState == INFECTED_SYMPTOMATIC then
                activity.probability ← min{activity.probability +
                    activity.deltaProbSick*(deltaCycle/deltaClock),1};
            else
                activity.probability ← min{activity.probability +
                    activity.deltaProbHealthy*(deltaCycle/deltaClock),1};
            end
        end
    end
end

```

Algoritmo 5.4: Regla de interacción para actualizar la probabilidad de las actividades.

5.1.5. Módulo de movilidad

El módulo de movilidad se encarga de transportar al agente hacia los lugares donde realiza sus actividades. Este contempla las variables internas que se enlistan a continuación:

- Bandera de movimiento: Para indicar que el agente se encuentra transportándose hacia un lugar se utiliza una variable booleana llamada `moving`. Esta es la misma señal que se encendía cuando el agente elegía un establecimiento para realizar una nueva actividad en la regla que se presentó en 5.6.

- Variables de velocidad del agente: Para señalar que ya se ha calculado un vector de velocidad para dirigir el agente hacia su lugar de destino se usa un booleano `trajectoryTraced`. Dicha velocidad se representa por un vector de dos dimensiones `velocity`, el cual se utiliza para mover al agente a una rapidez constante `speed` hacia su lugar de actividad.
- Radio de cercanía: Para determinar que el agente se encuentra ya suficientemente cerca del lugar al que busca llegar se usa un radio `Rp`. En el momento en que la distancia entre el agente y el lugar sea menor que dicha cantidad se establece que el agente ya no se encuentra en movimiento.
- Vector de posición: La posición del agente se guarda como un vector `position` de 2 componentes. Dicho vector se modificará con el vector de velocidad constante `velocity` hasta que el agente llegue a su destino.

Un diagrama de flujo de la regla de interacción que actualiza el movimiento del agente se muestra en la figura 5.8.

En forma resumida, cada vez que se requiera mover al agente hacia un lugar nuevo se calculará un vector de velocidad, cuya dirección vaya desde el lugar donde se encuentra el individuo hacia su destino. Usando dicho vector se irá actualizando la posición del agente, hasta el punto en que la distancia entre su posición y la del lugar sea menor que el radio de cercanía R_p . Una vez que esto pase se indicará que el agente ya no se encuentra en movimiento haciendo falsa la variable `moving` y se agregará el estado \mathcal{A} del agente al conjunto de personas que se hallan dentro del lugar de llegada.

5.1.6. Módulo de protección y políticas implementadas

El módulo de protección guarda variables que regulan a nivel de agentes individuales las distintas políticas de protección y resguardo ante una pandemia que se implementaron. Los escenarios contemplados fueron el uso de mascarilla, distanciamiento social, aislamiento, vacunación y una política llamada semáforo en la que se fusionaron las primeras 3 medidas mencionadas.

Las políticas principalmente actúan cambiando el valor de algunos de los parámetros que se han introducido en los módulos ya tratados. En la figura 5.9 se muestra una tabla donde se resume el funcionamiento de cada política (segunda columna) y las variables del modelo que pueden verse afectadas en cada caso (tercera columna).

En las secciones siguientes se explicará con mayor detalle la dinámica que se buscó simular con cada política.

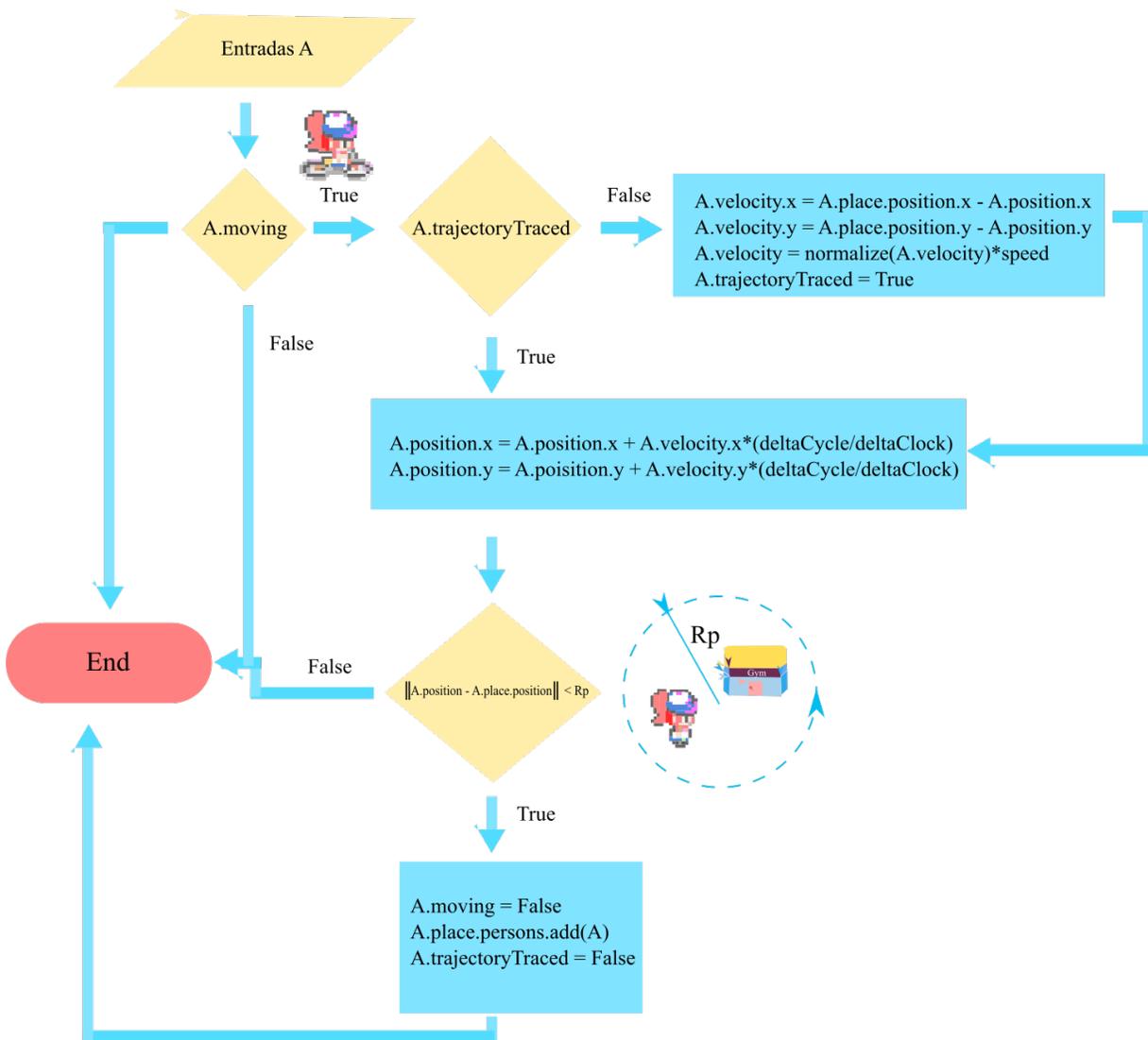


Figura 5.8: Regla de interacción para el movimiento de los agentes dentro de la ciudad.

Uso de mascarilla

La implementación del uso de mascarilla en esta tesis se basó en gran medida en el modelo consultado que se explicó en la sección 4.3.3. Esencialmente lo que se hace es disminuir la probabilidad de contagio α de los agentes susceptibles cuando estos o el agente infectado con el que tienen contacto se encuentran utilizando el cubrebocas.

Para indicar que el agente lleva puesta la mascarilla el módulo de protección utiliza una variable booleana llamada `usingMask`. El valor de esta para cada agente se decide cada vez que estos ingresan a un lugar o cuando comienzan a trasladarse por las calles de la ciudad. Dicha decisión de uso de cubrebocas utiliza distribuciones obtenidas con una encuesta, las cuales se detallarán en la sección 6.1.4. Para obtenerlas se les preguntó a los

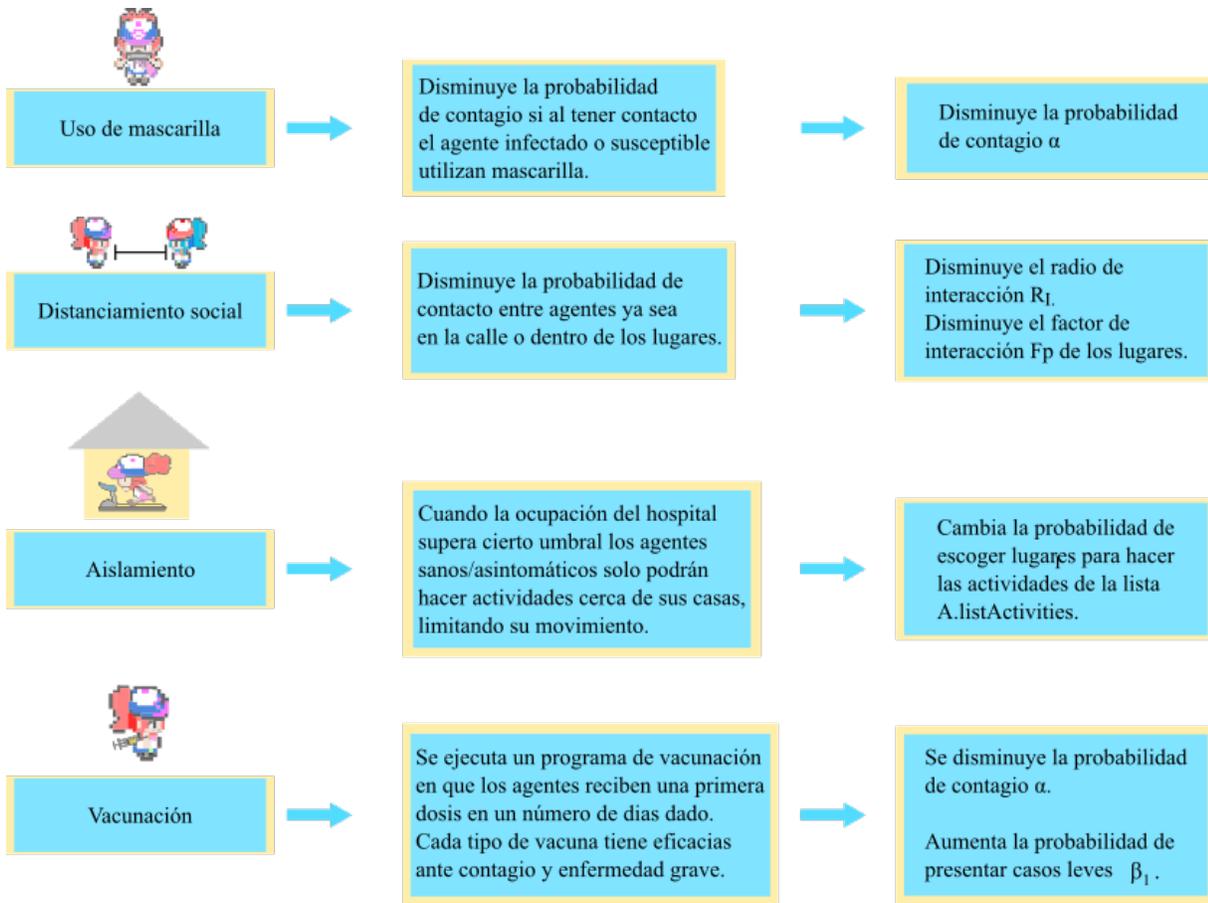


Figura 5.9: Efecto de las políticas implementados y los parámetros del modelo que se ven afectados por ellas.

participantes con que frecuencia solían utilizar el cubrebocas en distintos tipos de lugar y al movilizarse dentro de la ciudad donde habitaban.

La disminución de la probabilidad de contagio en esta política depende de si es el agente susceptible o el infectado el que lleva puesta la mascarilla. Así, cada vez que se de un contacto entre estos tipos de agente el primer paso a realizar será escoger una eficiencia de protección e_{mask} acorde a la tabla 5.1.

Susceptible usa mascarilla	Infectado usa mascarilla	Eficiencia e_{mask}
No	No	0
Si	No	0.85
No	Si	0.95
Si	Si	1

Tabla 5.1: Eficiencias a utilizar en la política de uso de mascarilla.

Una vez seleccionada la eficiencia correspondiente se disminuirá la probabilidad de

contagio para el agente susceptible con la siguiente ecuación:

$$\alpha_{\text{mask}} = \alpha(1 - e_{\text{mask}}) \quad (5.5)$$

Distanciamiento social

La política de distanciamiento social simula una situación en la que los agentes intentan mantener una distancia física mínima entre sí. El primer acercamiento propuesto para programar este comportamiento consistió en calcular fuerzas de repulsión entre los agentes, aplicándolas a su movimiento para que estos fueran alejándose entre sí. Dicho enfoque resultó sin embargo infructuoso, ya que era complicado lograr que los agentes fueran evitándose al mismo tiempo que se dirigían al lugar donde iban a realizar su actividad. Consultando en la literatura, se encontró que existen ciertos algoritmos propuestos para resolver este problema, como lo es Optimal Steps Model [22]. Este, sin embargo, involucra resolver en cada paso de la simulación un problema de optimización con campos de potencial, lo cual sobrepasa fácilmente los tiempos computacionales necesarios para lograr una simulación fluida. En efecto, los estudios que se han hecho utilizando dicho algoritmo suelen ocupar como máximo 100 agentes en las pruebas realizadas.

Ante la dificultad anterior se optó por simular en una forma simplificada pero eficiente el distanciamiento entre agentes. En concreto la solución que se implementó consistió en disminuir en una fracción e_{dst} tanto el radio de interacción R_I , como el factor de interacción F_p , utilizados para obtener el vecindario de los agentes (Véase el algoritmo 5.2). El valor disminuido de estas variables se calculó mediante las siguientes ecuaciones:

$$R_{I,\text{dst}} = R_I(1 - e_{\text{dst}}) \quad (5.6)$$

$$F_{p,\text{dst}} = F_p(1 - e_{\text{dst}}) \quad (5.7)$$

Estadísticamente las ecuaciones anteriores tienen un efecto de disminución en la cantidad promedio de agentes que habrá en el conjunto \mathcal{N} de vecinos de un agente dado, lo cual equivale en cierta medida a que las personas intenten alejarse entre sí. Una implementación similar de esta política, en la que se disminuía el radio de interacción se consultó en el modelo COVID - ABS explicado en la sección 4.3.1.

Aislamiento

En la política de aislamiento los agentes sanos o asintomáticos disminuyen su movilidad cuando la ocupación hospitalaria supera un umbral establecido. Dicho acercamiento fue inspirado en gran medida por el artículo presentado en la sección 4.3.4 en el que se restringía el traslado de personas a lugares lejanos dentro de una red de movilidad. En el modelo de esta tesis la restricción de movimiento se logra estableciendo que los agentes solo puedan hacer actividades en lugares que se encuentren alrededor de un radio R_A del lugar donde habitan.

Concretamente lo que se hace es una modificación de las distribuciones de probabilidad utilizadas para elegir los lugares donde realizan actividades las personas, en tal forma que se asigne una probabilidad cero a los lugares que se encuentran fuera del radio R_A .

Como se ha mencionado, la activación de esta política depende de que se sobrepase un umbral de ocupación hospitalaria que denotaremos como $h_{\text{isolation}}$. Una vez que esto ocurre se indica a los agentes sanos y asintomáticos que deben realizar una nueva actividad en sus casas, obligándolos así a retornar a su hogar. A partir de aquí se aplica la modificación descrita en el párrafo anterior limitando la movilidad de dichos habitantes a un radio cercano a donde habitan. Las restricciones permanecen activas por un número de días determinado $T_{\text{isolation}}$, tras lo que los individuos podrán volver a hacer sus actividades con normalidad.

Vacunación

La política de vacunación comienza asignando a cada agente una fecha en la que se les suministrará su primera vacuna, donde se reparten los días de aplicación en un periodo que abarca $T_{\text{vacunación}}$ días. La vacuna que recibe un agente se selecciona de entre las que se utilizaron en México, que fueron Pfizer, Astra Zeneca, Sputnik V, Sinovac, Cansino, Covaxin, Janssen y Moderna. Cada una determina el valor que se le da a 2 parámetros de protección: la eficacia ante contagio $e_{\text{vaccine},c}$ y la eficacia ante presentar un caso sintomático severo $e_{\text{vaccine},s}$. Así mismo, el tipo de vacuna determina el número de dosis D_{vaccine} que se requieren para tener la protección máxima. La asignación del tipo de vacuna se realizó una vez más utilizando una distribución obtenida con la encuesta que se presentará en la sección 6.1.4, donde se preguntó a las personas cual era el tipo de vacuna que habían recibido.

Las eficacias $e_{\text{vaccine},c}$ y $e_{\text{vaccine},s}$ se utilizan respectivamente para disminuir la probabilidad de contagio α y aumentar la probabilidad de contagiarse asintóticamente β_1 en la forma indicada por las ecuaciones 5.8 y 5.9.

$$\alpha_{\text{vaccines}} = \alpha \left(1 - \frac{N_{\text{vaccines}}}{D_{\text{vaccines}}} e_{\text{vaccines,c}} \right) \quad (5.8)$$

$$\beta_{1,\text{vaccines}} = \beta_1 + \frac{N_{\text{vaccines}}}{D_{\text{vaccines}}} e_{\text{vaccines,s}} (1 - \beta_1) \quad (5.9)$$

En dichas expresiones puede apreciarse que el efecto viene modulado por el número de dosis N_{vaccines} que haya recibido el agente hasta ese momento. En casos donde la vacuna requiera varias dosis D_{vaccines} , se van agendando las aplicaciones restantes en periodos subsecuentes de 30 días a partir de que el agente recibe su primera vacuna.

Semáforo

La política de semáforo combina 3 de los escenarios que se han explicado anteriormente: distanciamiento social, aislamiento y uso de mascarilla. Así mismo, considera distintos niveles de obediencia que podrían presentar los agentes para dichas medidas.

La probabilidad de obediencia se asigna con distribuciones obtenidas de la encuesta de calibración que se ha venido anunciando repetidamente. La pregunta de relevancia que se hizo en este caso fue qué tanto afectaba a los encuestados el semáforo epidemiológico implementado en México respecto a su uso del cubrebocas, distanciamiento social y aislamiento.

5.2. Clasificación del estado de salud a partir de datos de movilidad

En esta sección se detallan los pasos que se siguieron para estudiar la predicción del estado de salud de las personas partiendo de sus patrones individuales de movilidad.

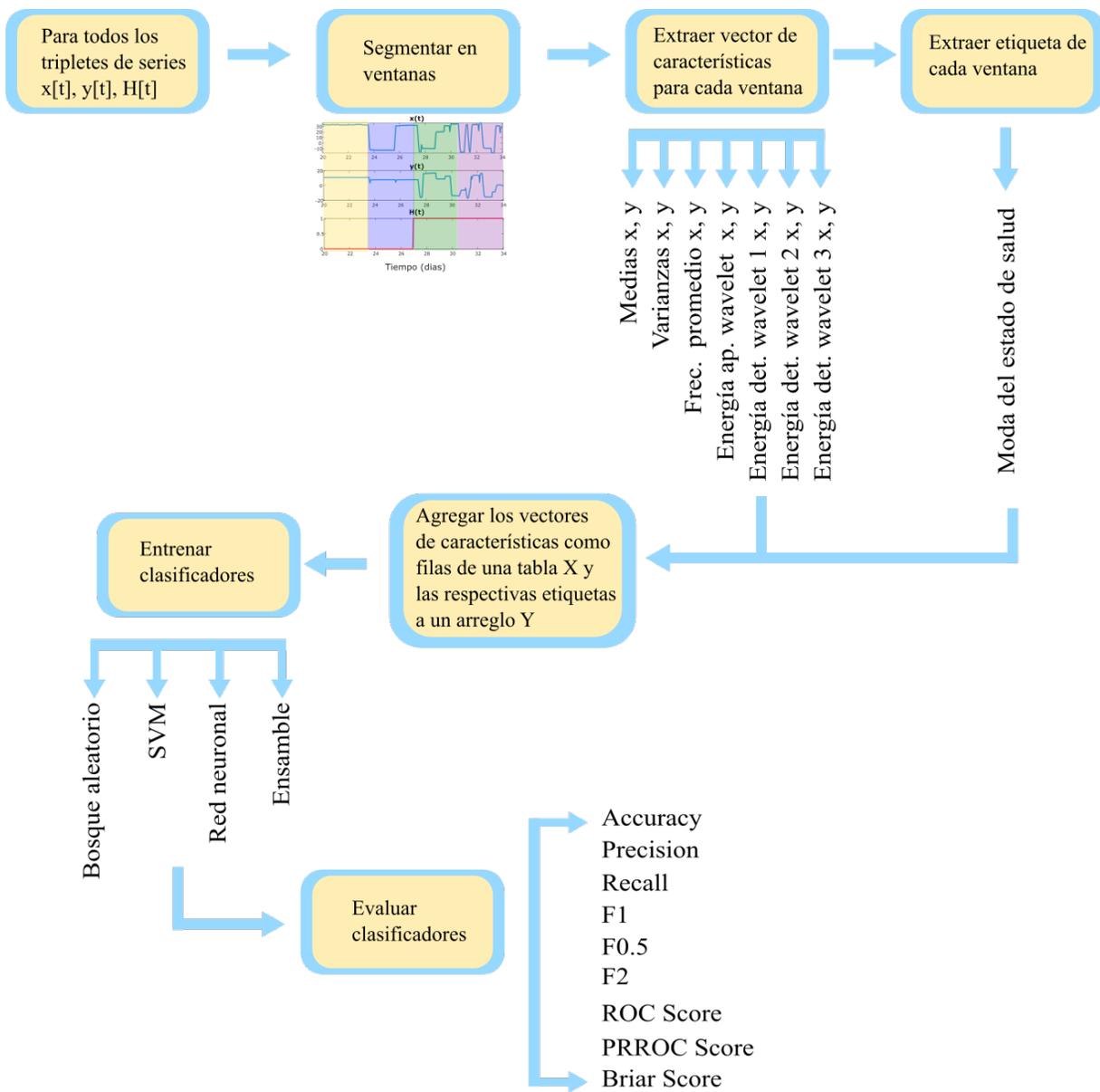


Figura 5.10: Pasos que se siguieron en la clasificación de los datos de movilidad.

Los datos que se utilizaron en esta parte del trabajo fueron series de tiempo de las coordenadas de posición $x[t], y[t]$ y estado de salud $H[t]$ de 30 individuos distintos, las cuales se generaron utilizando el modelo basado en agentes que se ha descrito en secciones anteriores.

El número total de agentes que contempla la simulación son 650, por lo que el porcentaje de agentes del que se obtuvieron datos fue aproximadamente 4%. La razón de haber utilizado un porcentaje pequeño de la población para el entrenamiento fue que la recolección de datos suponía una carga computacional importante. Sin embargo, el hecho de haber obtenido buenos resultados con dicha fracción de agentes sugiere que el método

pudiese ser viable en comunidades grandes aunque solo una pequeña parte de la población utilice la aplicación. Esto contrasta con otros métodos como el rastreo de contactos, que requieren un uso extensivo por parte de los habitantes.

En el caso del estado de salud es pertinente mencionar que se adaptaron los valores de la serie, de tal manera que solo se manejaran dos estados posibles: sano (0) y enfermo (1). Originalmente, como se mencionó en la sección 5.1.3, los agentes podían encontrarse en 6 estados de salud posibles: susceptible, expuesto, infectado asintomático, infectado sintomático, recuperado y fallecido, por lo que la serie de tiempo $H[t]$ constaba de valores que iban de 0 a 5. Lo que se hizo entonces fue reasignar dichos datos, en tal forma que en los lapsos donde el agente se hubiese encontrado infectado con síntomas se escribiera un uno, mientras que en el resto de los puntos se asignara un cero.

La forma en que se procesaron estas series se adhirió a la metodología de reconocimiento de actividad humana explicada en la sección 4.4. Dicho procesamiento se ha resumido en la figura 5.10.

En total, como se ha dicho, se tenían 30 tripletes de series de tiempo $x[t]$, $y[t]$, $H[t]$. Cada una de estas tercias fue procesada, para ir extrayendo vectores de características y etiquetas. El primer paso que se realizó con cada tercia fue un recorte de las series en ventanas de tiempo consecutivas. Denotaremos aquí como $x_i[t]$, $y_i[t]$ y $H_i[t]$ a los segmentos i -ésimos recortado de dichas series de tiempo. Una aclaración importante a realizar es que a pesar de que en la definición del problema de reconocimiento de actividad humana se mencionó que las ventanas no debían intersectarse, aquí dicha condición fue relajada, permitiendo que hubiese un porcentaje de traslape entre ellas. Lo anterior no solamente permitiría generar una cantidad mayor de vectores de características, sino que en cierta medida incorporaría una noción de memoria en los clasificadores entrenados, de forma que estos pudiesen detectar cuales valores solían preceder a otros dentro de las series de tiempo analizadas. Una vez recortados los segmentos, se obtuvieron para cada uno de ellos un vector de características y una etiqueta. El vector de características de dimensión 14, contenía 7 características obtenidas del recorte de la componente horizontal $x_i[t]$, y 7 de la vertical $y_i[t]$. Tales características son las que se enuncian a continuación:

- Media
- Varianza
- Frecuencia promedio
- Energía de la aproximación wavelet A_3 de la señal.

- Energía del primer detalle wavelet de la señal D_1 .
- Energía del segundo detalle wavelet de la señal D_2 .
- Energía del tercer detalle wavelet de la señal D_3 .

En este listado se tienen dos características que aportan información estadística de la serie de tiempo y cinco características que involucran una aplicación de técnicas de análisis en tiempo-frecuencia, como son la transformada de Fourier y la transformada wavelet. La razón de haber escogido estas últimas es que el movimiento de las personas suele seguir patrones periódicos bien establecidos (rutinas), y por tanto se determinó que la información aportada por características en los dominios de la frecuencia y wavelet resultaría importante. Tal suposición fue confirmada, además, con pruebas iniciales que se hicieron, en las que se observó que la frecuencia promedio y energías wavelet cambiaban cuando el agente se encontraba enfermo. Los resultados de dicha evaluación fueron los que se mostraron en las figuras 4.9 y 4.11 del marco teórico.

Una vez obtenido el vector de características de la ventana, se calculó la etiqueta que le correspondía. Para ello simplemente se obtuvo la moda del segmento de serie de tiempo del estado de salud $H_i[t]$, i.e. el valor de estado de salud que más hubiese predominado en el respectivo intervalo de tiempo.

Los vectores de características obtenidos para cada ventana fueron almacenándose como las filas de una matriz X , mientras que sus respectivas etiquetas se guardaron en un vector Y . Esta extracción de características, como se mencionó el principio, fue realizada para los tripletes $x[t], y[t], H[t]$, de cada uno de los 30 agentes.

La extracción de vectores características representa una especie de transformación, en la que las series de tiempo son llevadas a un espacio de características donde se facilita la identificación de puntos de tiempo donde la persona estuvo enferma. Ya que nos encontramos en dicho espacio, es importante, sin embargo, establecer cómo es que se van a separar los puntos de enfermedad de aquellos a etiquetar como sanos. Esta tarea, como se mencionó en el marco teórico la realizan modelos de clasificación, y es posible pensarla como una segunda transformación, que mapea el espacio de características a un espacio de etiquetas (sano(0) y enfermo(1)).

Los tipos de clasificador entrenados en este trabajo fueron 4: bosques aleatorios, máquinas de soporte vectorial, red neuronal, y un ensamble que combinaba la decisión de varios de ellos.

Antes del entrenamiento, sin embargo, se aplicó un paso de preprocesamiento importante a la tabla de vectores recolectados X . Dicho preprocesamiento consistió en normalizar

cada una de las columnas, de modo que los datos de cada característica se encontraran en escalas comparables ². El cálculo de normalización se realizó restando de cada dato x_{ij} de la tabla, la media de la columna en que se encontraba μ_j , y dividiendo entre la desviación estándar de la misma columna σ_j .

$$x_{ij} \leftarrow \frac{(x_{ij} - \mu_j)}{\sigma_j} \quad (5.10)$$

Una vez realizada esta normalización se procedió al entrenamiento de los clasificadores. Para cada uno de ellos se seleccionó aleatoriamente un subconjunto de vectores X_T con los que se entrenaría, donde el número de vectores contenidos en esta subtabla representaba un 80 % del total de vectores en el conjunto original X . Las etiquetas correspondientes al conjunto de entrenamiento fueron guardadas en un arreglo Y_T .

En total se entrenaron 2 máquinas de soporte vectorial, 2 bosques aleatorios, 1 red neuronal y un ensamble que agrupaba los 5 anteriores. Ya hecho esto, se evaluó cada uno de los clasificadores utilizando el 20 % de vectores que hubiesen sobrado de aquellos que se escogieron para entrenar. A los vectores y etiquetas de dicho conjunto de validación se les denotará aquí como X_v, Y_v . La validación de un clasificador involucra el cálculo de métricas, que proporcionen información sobre que tan acertado es el modelo cuando etiqueta datos que no se le habían presentado como parte de su entrenamiento. Una de las medidas que más suelen utilizarse para este propósito es la exactitud (accuracy), en la que se cuenta la proporción de verdaderos positivos y verdaderos negativos respecto al total de datos con los que se valida. Los verdaderos positivos representan sencillamente, la cantidad de veces que el clasificador asignó correctamente la etiqueta 1 para los vectores de X_v , mientras que los verdaderos negativos será la cantidad de veces que se asignó correctamente la etiqueta 0 dentro de dicho conjunto. Es necesario, sin embargo, tener precaución, ya que existen ciertas instancias donde la métrica antes mencionada puede resultar engañosa. Supóngase por ejemplo, que se tiene un conjunto de datos en el que la proporción de vectores con la etiqueta 0 constituye 95 % del total de datos. Uno podría en principio dictar al clasificador que asigne el valor 0 a absolutamente cualquier vector que se le presente y su accuracy sería relativamente alto (0.95), ya que el porcentaje de datos con la etiqueta 0 es grande. En estos casos, en los que se tiene un desbalance de clases puede ser más conveniente recurrir a otro tipo de métricas, donde se dé prioridad a la detección de una etiqueta sobre la otra.

²Este suele ser un paso importante en la clasificación de datos que provienen de distintas escalas. Por ejemplo, en este caso permitiría que la predicción del estado de salud funcione independientemente del tamaño de la ciudad donde habiten las personas y/o el rango de movimiento de cada individuo. Una cuestión importante que permanece, sin embargo, es si dicha escala afecta en forma importante a los patrones de propagación de enfermedad y por tanto si hacer la normalización hace que se pierda información relevante para conocer el estado de salud de una persona.

Si lo pensamos, esta realmente es la situación en la que nos hallamos, ya que la porción de tiempo en que un individuo se encuentra sano es relativamente alta si se le compara con los periodos en que se encuentra enfermo, y por tanto los vectores de características que corresponderán a un estado de infección constituyen un porcentaje pequeño del total de datos.

Las métricas adicionales que suelen usarse para evaluar un clasificador en casos de desbalance de clases son las siguientes:

- **Precision:** Es la razón de verdaderos positivos respecto al total de positivos detectados por el modelo. Cuando esta cantidad es baja significa que el modelo entrenado genera muchos falsos positivos.
- **Recall:** Es la razón de verdaderos positivos respecto al total de positivos reales que debieron detectarse. Si esta cantidad es baja significa que se asignan muchos falsos negativos.
- **F0.5:** Esta métrica resulta de ponderar Precision y recall, en tal forma que se de más peso en la evaluación a que el clasificador no genere falsos positivos. Su definición es la siguiente:

$$F_{0.5} = \frac{((1 + 0.5^2)\text{Precision} \times \text{Recall})}{(0.5)^2\text{Precision} + \text{Recall}} \quad (5.11)$$

- **F1:** Esta cantidad también pondera Precision y Recall, evaluando con igual peso que el modelo no genere falsos positivos y falsos negativos. Su definición es:

$$F = \frac{2\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.12)$$

- **F2:** La última ponderación a usar de Precision y Recall evaluará con mayor peso que el clasificador no genere falsos negativos, es decir que siempre logre asignar bien la etiqueta 1 aunque a veces se equivoque al hacerlo. La expresión que define a esta cantidad es:

$$F_2 = \frac{(1 + (2)^2)\text{Precision} \times \text{Recall}}{(2)^2\text{Precision} + \text{Recall}} \quad (5.13)$$

- **Área bajo la curva ROC:** Una curva ROC gráfica la tasa de verdaderos positivos contra la tasa de falsos positivos para distintos umbrales de decisión de un clasificador. Para entender dicha definición es necesario mencionar que los clasificadores

pueden no arrojar directamente la etiqueta a asignar para un vector de características dado, sino que podrían antes calcular la probabilidad de asignar cada etiqueta. Para el caso concreto en que hay 2 etiquetas se puede por lo tanto calcular la probabilidad de asignar 1 y si esta es mayor a un umbral de 0.5 arrojar como resultado dicha categoría. En la construcción de una curva ROC dicho umbral de decisión, que por default es 0.5 va modificándose desde 0 hasta 1, y para cada uno de los valores se obtienen las tasas de verdaderos positivos y falsos positivos que genera el modelo con los datos de evaluación. El área bajo la curva generada al graficar dichas tasas representa una métrica que evalúa la correcta asignación tanto de la etiqueta 0 como de la etiqueta 1.

- **Área bajo la curva Precision - Recall (PR):** Una curva similar a ROC, en la que se da prioridad a la correcta asignación de la etiqueta 1 sobre la etiqueta 0 es la curva Precision - Recall. Para construir esta lo que se hace es variar el umbral de decisión del clasificador y para cada valor del umbral calcular Precision y Recall con el conjunto de datos de evaluación. El área bajo la curva que resulta de graficar estas cantidades puede usarse posteriormente como una métrica, que evalúa lo acertado que es el clasificador al asignar la etiqueta positiva.
- **Briar Score:** Finalmente Briar Score representa el error cuadrático medio de la probabilidad arrojada por el clasificador para la asignación de la etiqueta 1. Si el conjunto de evaluación consta de N vectores de características, la probabilidad de asignar la etiqueta 1 al vector i -ésimo es y_i y la etiqueta real que corresponde a dicho vector es \hat{y}_i , Briar Score se calculará de la siguiente manera:

$$\text{Briar Score} = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2 \quad (5.14)$$

Las medidas antes listadas fueron las que se utilizaron para evaluar los distintos clasificadores que se entrenaron.

Con ello se concluye la explicación de la metodología que se siguió para plantear el modelo de agentes desarrollado y para realizar las tareas de clasificación que se aplicaron a los datos generados con dicho modelo.

A lo largo de estas secciones se han enunciado varios parámetros, cuyo valor puede determinar en forma importante los resultados que se obtengan. El siguiente paso a realizar consiste, por tanto, en asignar a dichos parámetros valores adecuados, de tal forma que nuestro estudio computacional sea equivalente en cierto sentido a investigar el fenómeno

real que nos concierne: la movilidad en una pandemia.

Calibración y ajuste de parámetros

En esta sección se explicará el modo en que se ajustaron los parámetros que contempla la metodología de esta tesis. Respetando la cronología que se ha usado en capítulos anteriores, se describirá en primer lugar como se establecieron los parámetros del modelo basado en agentes desarrollado, para posteriormente expresar el valor de los parámetros usados en la clasificación de datos de movilidad.

6.1. Modelo basado en agentes

En el modelo basado en agentes fue necesario ajustar los parámetros epidemiológicos utilizados para actualizar el estado de salud de los individuos (Figura 5.5) y también las distribuciones de probabilidad para la duración, frecuencia y lugares de realización de las distintas actividades que llevaban a cabo los agentes como parte de su rutina. El modo en que se ajustaron estos será detallado en los apartados siguientes.

6.1.1. Ajuste de parámetros epidemiológicos

El primer parámetro epidemiológico que se estableció fue la probabilidad α de que un susceptible se contagiara al tener contacto con un infectado. En este trabajo, la enfermedad que se buscó simular concretamente fue COVID - 19, y por tanto se consultó en la literatura cuál era el porcentaje de contagio del coronavirus SARS-CoV-2 cuando se tenía contacto con una persona enferma. A este respecto se hallaron algunos estudios que reportaban probabilidades por encima del 90% para personas altamente contagiosas. En última instancia el valor que se le dio a α fue 0.98 [23].

Los siguientes parámetros que se ajustaron fueron β , γ y ω , que en el modelo representan probabilidades de pasar desde una fase de la enfermedad a la siguiente en cada ciclo de la simulación. Estas por supuesto no es posible asociarlas directamente a datos reales,

y por tanto fue necesario definir otros parámetros auxiliares β_d , γ_d y ω_d , que representarían la probabilidad por día de pasar desde un estadio de la enfermedad al siguiente. El cálculo por ejemplo de β a partir de su auxiliar correspondiente β_d se puede hacer utilizando el factor de conversión de tiempo que se enunció en 5.1. En este caso específico se haría:

$$\beta = \beta_d \left(\frac{1}{24h} \right) \left(\frac{\Delta_{\text{cycle}}}{\delta_{\text{clock}}} \right) \quad (6.1)$$

donde Δ_{cycle} representa el tiempo en segundos que tomó el último ciclo de actualización de la simulación.

Para calibrar β_d se consultó información sobre el número de días que solía tomar el periodo de incubación de COVID - 19. Acorde a la CDC, la aparición de síntomas puede llegar a tardar 14 días [24], por lo que se fijó un valor para este parámetro de $\beta_d = 1/14d = 0.0714d^{-1}$. Por otro lado, para γ_d y ω_d , se investigaron las duraciones promedio de cuadros infecciosos con COVID - 19 para pacientes asintomáticos y sintomáticos respectivamente. Existen publicaciones donde se ha reportado un valor promedio de 10 días para este tiempo en pacientes asintomáticos o con signos leves y hasta 30 días en casos serios [25][26], de modo que se fijaron $\gamma_d = 1/10d = 0.1d^{-1}$ y $\omega_d = 1/30d = 0.03d^{-1}$.

Otras cantidades se calibraron directamente con datos encontrados en la literatura. Para β_1 se usó la probabilidad de presentar un caso asintomático, cuyo valor se ha estimado en 0.13 [27], mientras que para γ_1 y ω_1 se consultó la probabilidad de generar inmunidad después de una infección con COVID - 19, que se ha reportado en 0.9 [28]. Finalmente, ω_2 se calibró usando la probabilidad de fallecer al contraer la enfermedad, que en México era de aproximadamente 5% antes de que se aplicaran las vacunas [29].

Una tabla que recopila los valores asignados para los parámetros epidemiológicos antes mencionados, y otros que se ajustaron empíricamente se muestra a continuación:

Parámetro	Descripción	Valor
α	Probabilidad de contagio al contactar un infectado.	0.98
β_d	Probabilidad por día de salir del periodo de incubación.	0.0714
γ_d ,	Probabilidad por día de salir un cuadro de infección asintomática.	0.1
ω_d	Probabilidad por día de salir de un cuadro de infección seria	0.03
β_1	Probabilidad de presentar un caso asintomático	0.13
γ_1, ω_1	Probabilidad de generar inmunidad después de infección.	0.9
ω_2	Probabilidad de fallecer al contraer después de infección.	0.05
μ_0	Tasa de generación de nuevas variantes (Ec. 5.3).	0, 1, 3
C_{sat}	Constante de saturación (Ec. 5.3).	0, 5, 10

Tabla 6.1: Valor asignado a los parámetros epidemiológicos utilizados dentro del modelo basado en agentes desarrollado.

En el caso de los parámetros de mutación se utilizó un valor 0 para la mayor parte de pruebas realizadas. Una demostración de qué es lo que ocurre cuando se activa la generación de nuevas variantes se hará en la sección 7.2; Para dichos resultados se usaron valores de 1 y 3 para μ_0 y de 5 y 10 para C_{sat} .

6.1.2. Calibración de las actividades de los agentes utilizando una encuesta

Como se mencionó en partes de la metodología, la duración, frecuencia y tipos de lugar asociados a cada una de las actividades que pueden hacer los agentes se muestrearon usando distribuciones obtenidas con información de una encuesta.

Dicha encuesta de actividades fue realizada en un periodo que abarcó desde el 28 de Septiembre al 10 de Octubre del año 2021, recopilando un total de 123 respuestas. El método de muestreo utilizado para elegir a los participantes fue el de bola de nieve. Este consiste en compartir en primera instancia el cuestionario a conocidos cercanos, pidiéndoles que de favor la difundan a su vez con otras personas. Dicho proceso se repite con cada nuevo encuestado, logrando así obtener respuestas en una red de contactos que va haciéndose cada vez más amplia.

Respecto a las características demográficas de los participantes se obtuvo que la mayoría de ellos residían en la Ciudad de México y tenían una edad entre 20 y 30 años. El sexo de los encuestados permaneció relativamente balanceado, obteniéndose porcentajes femeninos y masculinos de participación de 52.8% y 47.2% respectivamente. En la figura 6.1, se muestran las distribuciones de probabilidad del estado de residencia, edad y sexo de las personas que respondieron el cuestionario.

El cuestionario como tal se dividió en dos partes. La primera de ellas se destinó a recopilar información sobre la frecuencia con que las personas hacían las actividades consideradas dentro del modelo de agentes, la duración que solía llevarles realizarlas, y el tipo de lugares donde habitualmente las llevaban a cabo. Esta es la parte que se describirá en esta sección. En la segunda parte del cuestionario se hicieron preguntas relacionadas con medidas de protección a la pandemia (uso de cubrebocas, aislamiento, distanciamiento social, vacunación y semáforo epidemiológico), y la información obtenida se utilizó para calibrar las distintas políticas implementadas en el modelo, como se detallará en la siguiente sección.

Como se mencionó en 5.1.4 las actividades modeladas fueron: comer, dormir, trabajar, comprar comida, ejercitarse, ir al médico y socializar (hacer actividades recreativas). Las distribuciones de frecuencia, duración y tipos de lugar obtenidas para cada una de estas

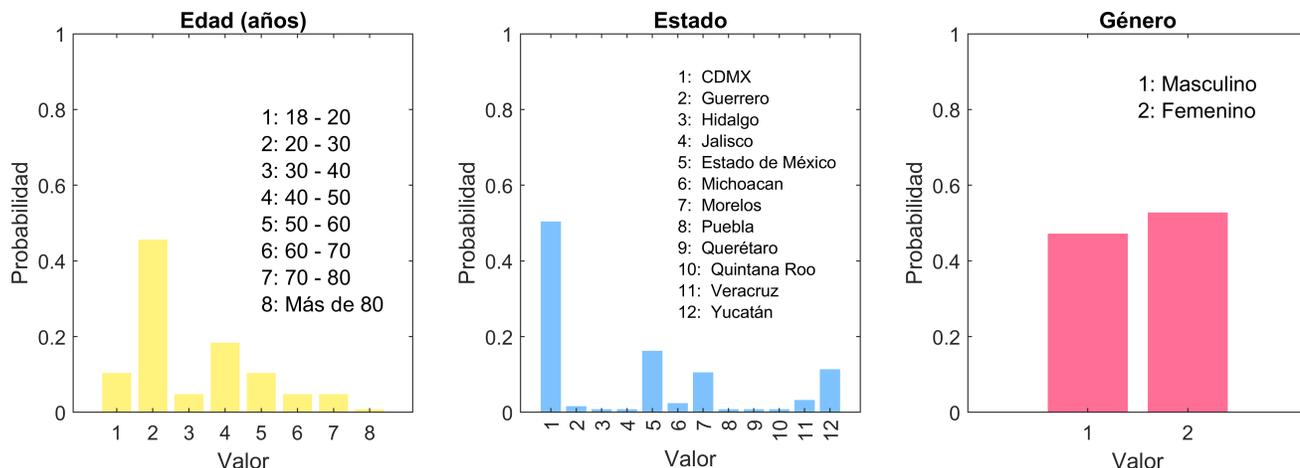
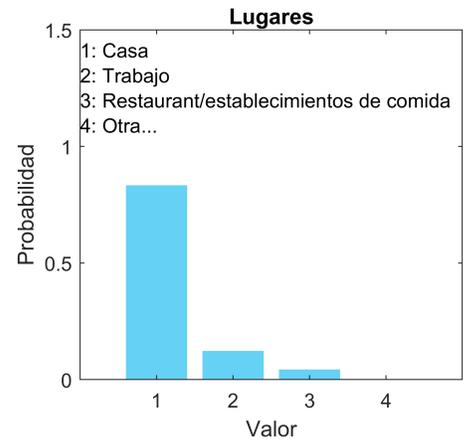
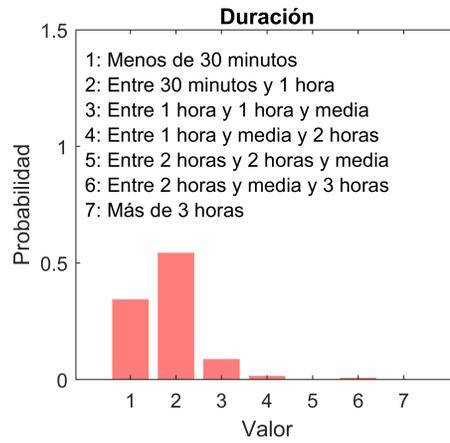
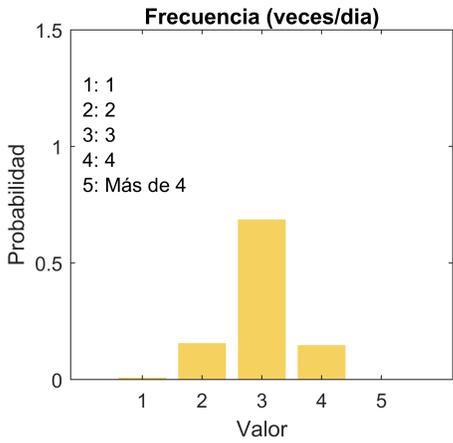
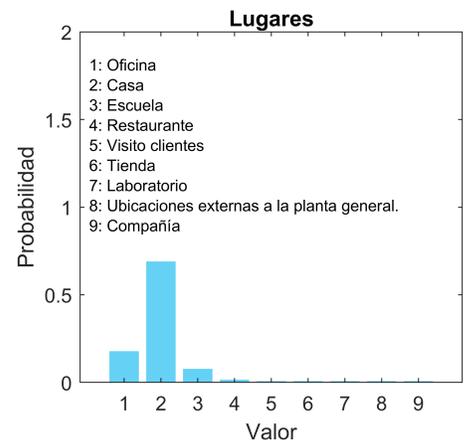
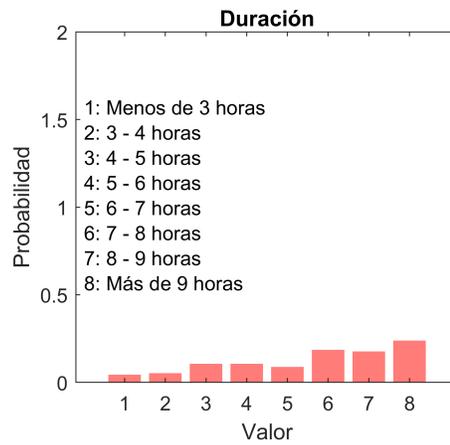
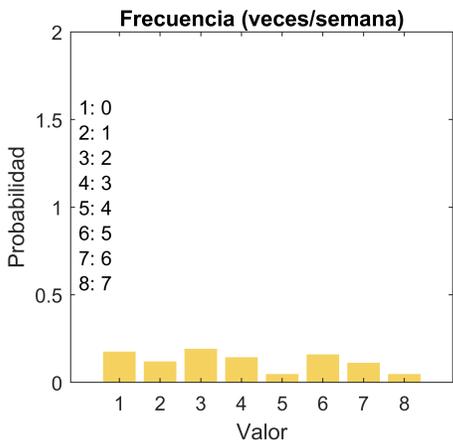


Figura 6.1: Distribuciones de edad, estado de residencia y género de los participantes que contestaron la encuesta utilizada para calibrar las actividades de los agentes.

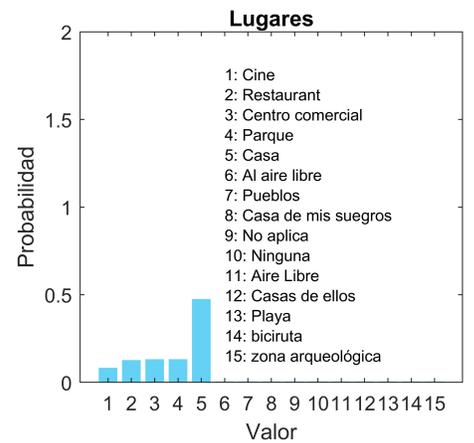
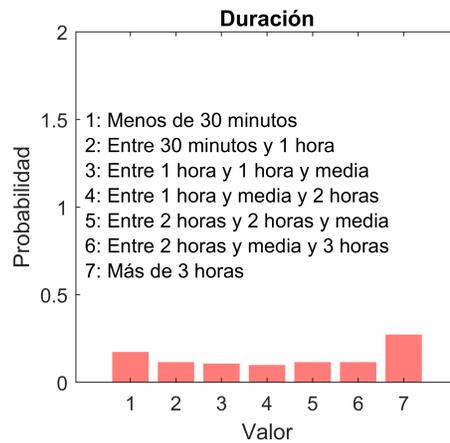
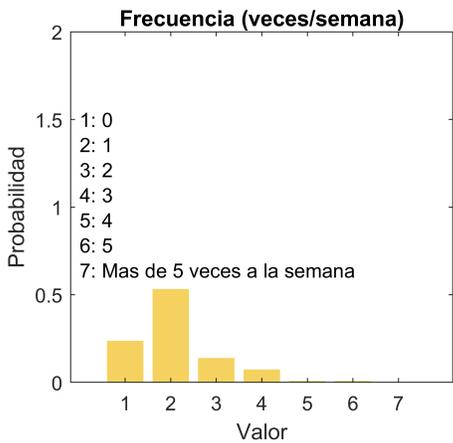
tareas son las que se muestran en la figura 6.2. En el caso de la actividad "ir al médico" se hizo la suposición de que esta conllevaba necesariamente acudir a un centro de atención hospitalaria o un recinto análogo, y por tal razón no se adquirió información sobre el tipo de lugares al que acudían las personas para hacer tal actividad. Otra suposición importante que se hizo fue que la actividad "dormir" era realizada una vez al día por las personas y siempre en su hogar. Por tal razón para ella solo se obtuvo la distribución de duración, consultando cuantas horas al día solían dormir aproximadamente los individuos. Finalmente, para culminar con aclaraciones importantes a mencionar sobre la encuesta, en la actividad comprar comida se supuso que esta siempre se realizaba en un supermercado y por ello solo se obtuvieron datos de su duración y frecuencia.



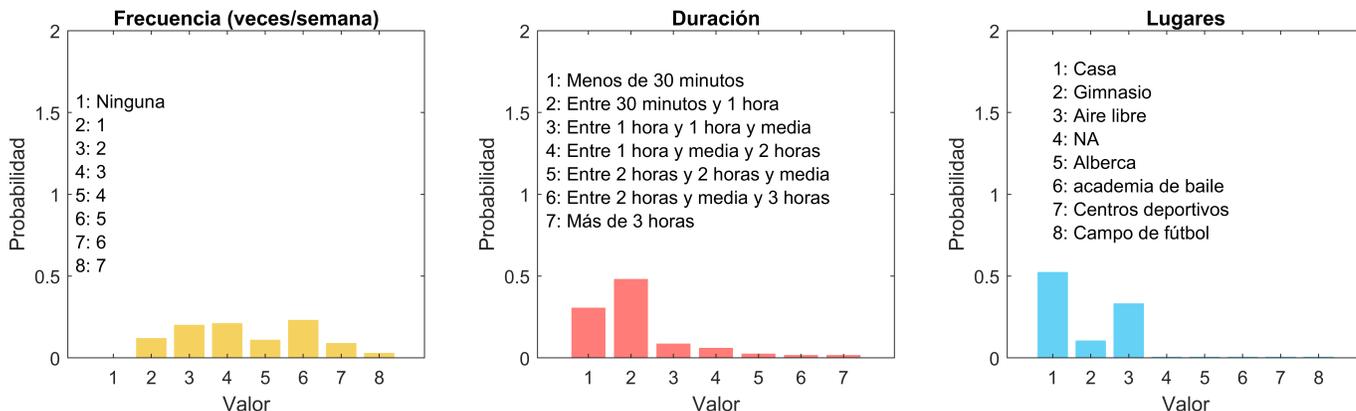
(a) Comer



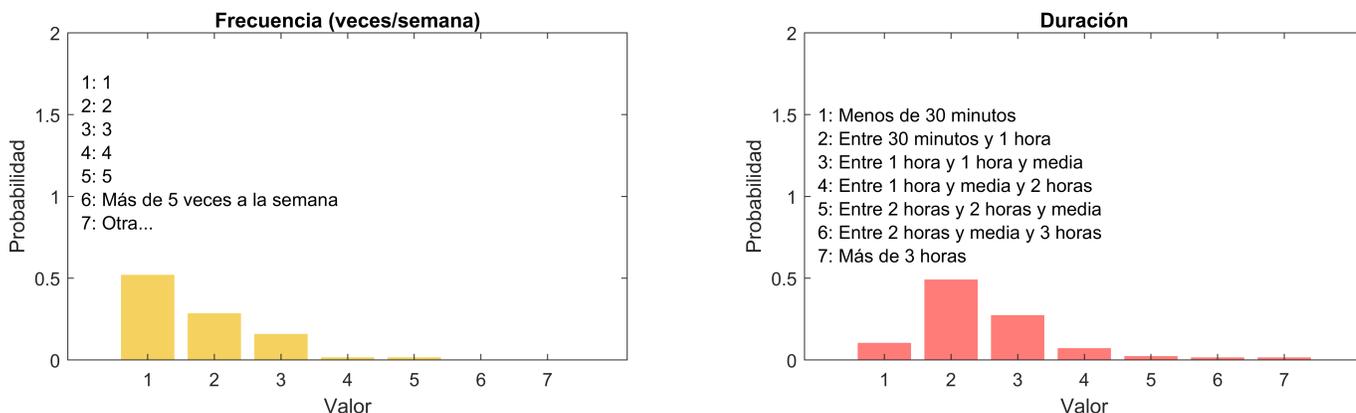
(b) Trabajar



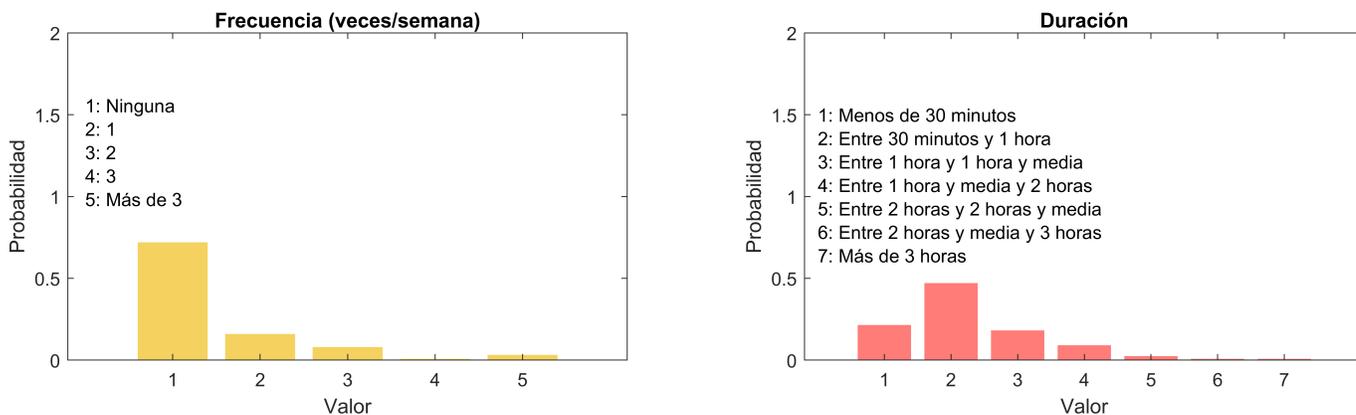
(c) Socializar (actividades recreativas)



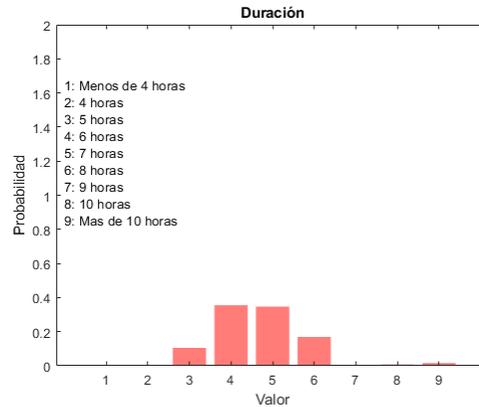
(d) Ejercitarse



(e) Comprar comida



(f) Ir al médico



(g) Dormir

Figura 6.2: Distribuciones de frecuencia, duración y duración de las distintas actividades consideradas en el modelo de agentes, construidas usando información del cuestionario aplicado.

Dentro del modelo de agentes las frecuencias se manejan por hora y es por ello que los valores de algunas de las distribuciones que se han mostrado tuvieron que ser reescaladas adecuadamente antes de ser incorporadas dentro del programa. Por ejemplo, para el caso de comprar comida las opciones que se dieron a los participantes en la encuesta venían dadas en unidades de 1/semana, y por tanto fue necesario convertir dichos valores multiplicándolos por el factor de conversión $\left(\frac{1\text{semana}}{7\text{días}}\right) \left(\frac{1\text{día}}{24h}\right)$.

Otro ajuste que se hizo fue que en aquellos casos donde se hubiese dado como opción la respuesta "más de X ", con X un número entero, se asignó dentro del programa un valor fijo de $X + 1$ para dicha categoría.

Finalmente, se tuvo la necesidad de combinar algunas categorías para ciertas distribuciones. Esto ocurrió sobre todo en el caso de las distribuciones de los tipos de lugar, ya que en el cuestionario aplicado se dio la opción de que los participantes escribieran respuestas nuevas si es que no venía el lugar donde ellos solían realizar la actividad en cuestión. Como se explicó en la sección 5.1.1 la ciudad virtual construida contenía ya una serie de tipos de lugar predefinidos, y por tanto fue necesario encapsular las opciones nuevas incluidas por las personas dentro de ellos. En el caso por ejemplo, de la actividad ejercitarse algunos de los encuestados agregaron categorías como "Alberca", "Academia de baile", "Centros deportivos" y "Campo de fútbol". Todas ellas fueron combinadas dentro de la opción Gimnasio en un análisis posterior de la encuesta, para que se pudiera incorporar correctamente dentro del modelo de agentes implementado.

6.1.3. Calibración del número de agentes

Un parámetro importante a determinar fue el número de agentes N que habitarían en el mundo virtual. En el programa implementado, la eficiencia computacional viene limitada principalmente por el algoritmo de obtención de vecindarios de los agentes, cuya complejidad es cuadrática respecto al número de agentes. En efecto, si se hace un análisis cuidadoso de los algoritmos y reglas de interacción del modelo de agentes se llegará a la conclusión de que el tiempo de actualización T_{cycle} de cada ciclo en nuestro modelo hereda dicha complejidad cuadrática:

$$T_{\text{cycle}} \in \mathcal{O}(N^2) \quad (6.2)$$

El número de agentes a utilizar dependerá del poder computacional con el que se cuente para ejecutar el programa. En nuestro caso se trabajó principalmente con un procesador Intel Core i5 de octava generación, con una frecuencia de operación de 1.6 GHz. Bajo dicho valor, el número de agentes que podrían utilizarse sin exceder un tiempo de 0.01s en el ciclo de actualización del programa, suponiendo además un procesamiento de 32 bits en cada ciclo sería el siguiente:

$$\begin{aligned} N^2 &= (1.6 \times 10^9) \times 0.01s/32 = 500000 \\ \implies N &= \sqrt{500000} \approx 707 \end{aligned} \quad (6.3)$$

El cálculo anterior por supuesto constituye una aproximación, ya que las suposiciones estrictamente hablando no se cumplen para todos los ciclos de reloj y un análisis más riguroso debería tomar en cuenta otros aspectos del modo en que opera un procesador. Por ello, lo que se hizo fue tomar dicho número como referencia y hacer pruebas con cantidades de agentes relativamente cercanas a dicho valor. En última instancia se encontró que se conseguía una simulación fluida con 650 agentes, por lo que este fue el número de habitantes utilizado en las pruebas de este trabajo.

6.1.4. Ajuste de parámetros en las políticas de protección y resguardo

Como se mencionó antes, en la encuesta realizada también se hicieron preguntas destinadas a ajustar algunos parámetros dentro de las políticas de protección y resguardo implementadas. Concretamente, las políticas que se calibraron de dicha forma fueron el

uso de cubrebocas, la vacunación y la política de semáforo, en la que se combinan 3 medidas distintas: aislamiento, distanciamiento social y cubrebocas.

Para la política de cubrebocas se adquirió información sobre la frecuencia con que las personas solían utilizar la mascarilla en distintos tipos de lugar y cuando se movilizaban por la ciudad. Dichas distribuciones son las que se muestran en la figura 6.3. Estas fueron usadas dentro del modelo para determinar si un agente usaría mascarilla al ingresar a un nuevo lugar, y también cuando tuviera que moverse por la ciudad.

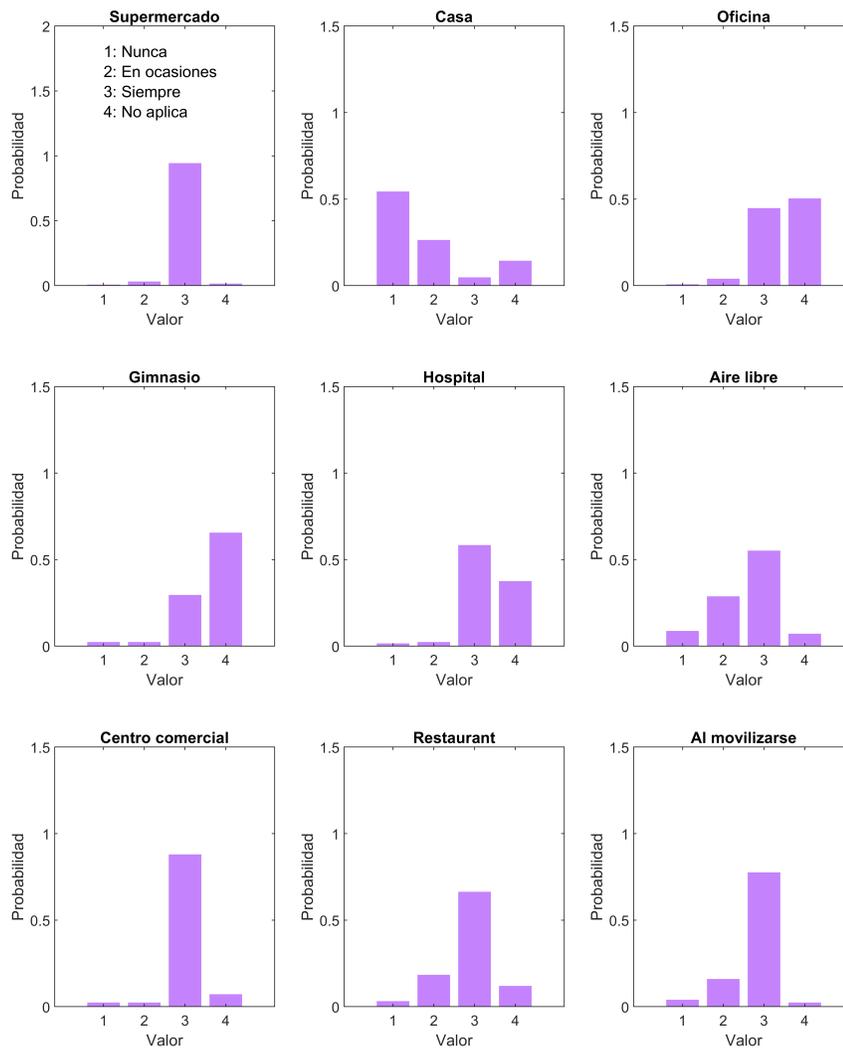


Figura 6.3: Distribuciones de uso de cubrebocas en los distintos tipos de lugar considerados en el modelo de agentes, construidas con información de una encuesta.

Por otra parte, para la política de vacunación se adquirió una distribución de las distintas vacunas que habían recibido hasta ese momento los encuestados. Dicha distribución, que se muestra en la figura 6.4 se utilizó en el modelo para muestrear la vacuna que se aplicaría a cada uno de los agentes.

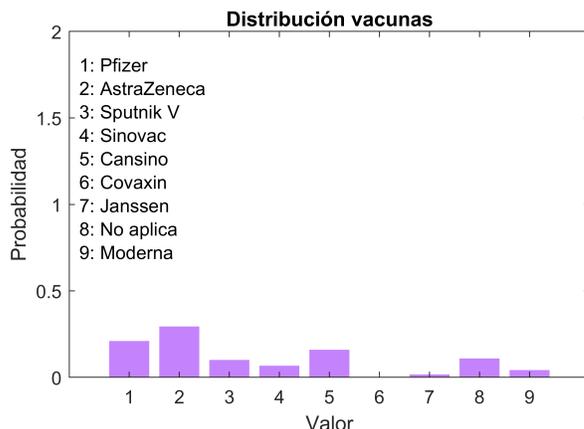


Figura 6.4: Distribución de vacunas aplicadas en México (Información obtenida del 28 de Septiembre al 10 de Octubre del año 2021).

Finalmente para la política de semáforo se preguntó a los encuestados en qué magnitud afectaba la instauración de un semáforo de alerta epidemiológica en su uso del cubrebocas, disminución de salidas y distanciamiento de otras personas. Las distribuciones obtenidas de las preguntas realizadas, que se muestran en la figura 6.5, fueron usadas para determinar la probabilidad de obediencia de los agentes ante las distintas restricciones que abarcaba la política de semáforo.

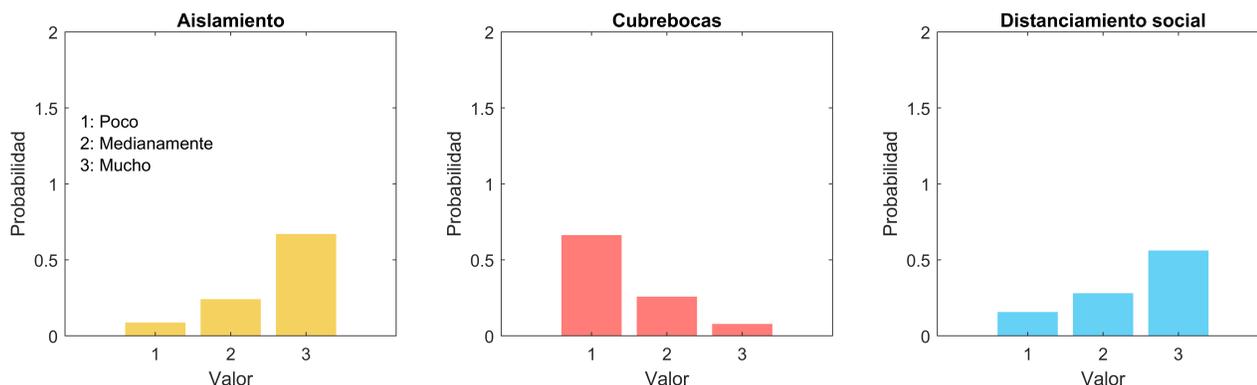


Figura 6.5: Distribuciones de la influencia que tiene el semáforo epidemiológico en los individuos encuestados.

6.2. Clasificación de datos de movilidad

En esta sección se presentarán en forma breve los parámetros utilizados en las tareas de clasificación que se aplicaron a los datos de movilidad que se exportaron del modelo de agentes.

6.2.1. Parámetros utilizados en la segmentación de series de tiempo y extracción de características

Como se mencionó en la sección 5.2, las series de tiempo de las coordenadas de posición de los agentes fueron segmentadas en ventanas, y para cada uno de los trozos se extrajeron distintas características. Los primeros parámetros que se establecieron al realizar dicha tarea fueron el tamaño de ventana, fijado en 50 puntos de la serie, y el porcentaje de traslape que se iba a repetir de una ventana a la siguiente, al que se le dio un valor de 90%. La razón de que se eligiera un porcentaje relativamente alto es que se quería que los clasificadores a utilizar adquirieran cierta memoria y fueran capaces de vislumbrar cambios respecto a puntos anteriores en las series de tiempo.

Dentro de las características que se extrajeron para cada ventana de tiempo se encontraban algunas que requerían realizar un análisis multiresolución de la señal con Wavelets, descomponiéndola en aproximación y detalles (Véase la sección 4.4.1). Uno de los elementos que fue necesario escoger al respecto fue la Wavelet madre a utilizar. Aquí en realidad se decidió usar aquella que nos resultaba más familiar, y que habíamos construido a pie en la computadora en nuestro proceso de aprendizaje sobre la transformada Wavelet. Esta fue la Wavelet madre Daubechies de orden 3, cuyo gráfico se muestra en la figura 6.6, y cuyos filtros pasa altas h y pasa bajos g usados en la descomposición de la señal son los que siguen:

$$h = \frac{1}{\sqrt{2}} \left[-\frac{(1 + \sqrt{3})}{4}, \frac{(3 + \sqrt{3})}{4}, -\frac{3 - \sqrt{3}}{4}, \frac{1 - \sqrt{3}}{4} \right] \quad (6.4)$$

$$g = \frac{1}{\sqrt{2}} \left[\frac{(1 - \sqrt{3})}{4}, \frac{(3 - \sqrt{3})}{4}, \frac{(3 + \sqrt{3})}{4}, \frac{(1 + \sqrt{3})}{4} \right] \quad (6.5)$$

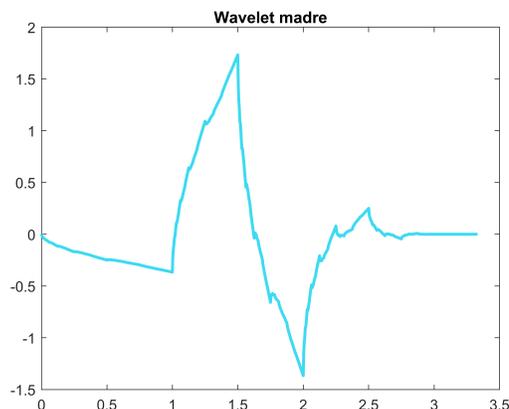


Figura 6.6: Madre Wavelet Daubechies - 4 utilizada en la extracción de características en el dominio Wavelet.

6.2.2. Parámetros utilizados en los clasificadores

Ya explicados los parámetros usados en la extracción de características, resta únicamente indicar aquellos que se usaron en los modelos de clasificación que se entrenaron. Como se dijo dentro de la explicación teórica provista para ellos en 4.4.2, cada modelo de clasificación cuenta con varios detalles que es necesario especificar. En la tabla 6.2 se ha hecho un compendio del diseño que se le dio a los clasificadores utilizados en este trabajo.

SVM	
Kernel	Gaussiano
Bosques aleatorios	
N. árboles	20
Profundidad máxima	10
Criterio de separación	Ganancia de información
Red neuronal	
Neuronas en capa intermedia	20
Función de activación intermedia	log - sigmoide
Función de activación final	Softmax

Tabla 6.2: Parámetros utilizados para entrenar los distintos clasificadores

Con esto termina la descripción del modo en que se calibraron los distintos parámetros del modelo basado en agentes y las tareas de clasificación. El siguiente capítulo se dedicará a la presentación y discusión de los resultados obtenidos en esta tesis.

Resultados y discusión

En esta sección se presentarán y discutirán los distintos resultados que se obtuvieron en este trabajo. Se iniciará con una recopilación de las series de tiempo macroscópicas que arrojó el modelo de agentes para los distintos escenarios de protección implementados, comparándolas y discutiendo cuál de las políticas podría proveer un mejor control en la propagación de la pandemia. Hecho esto, se hará un compendio de los resultados obtenidos en la clasificación del estado de salud de personas individuales usando sus datos de movimiento. Dicha parte se enfocará sobre todo en discutir si es viable hacer la predicción del estado de salud con las herramientas propuestas, y si dicha predicción se vuelve menos certera cuando se imponen políticas que restringen el movimiento de las personas, como es el aislamiento.

7.1. Modelo basado en agentes

7.1.1. Series de tiempo SEIRD

En la figura 7.1 se ilustran los gráficos de evolución de las fracciones de susceptibles, expuestos, infectados, recuperados y fallecidos que se obtuvieron en los distintos escenarios de protección y resguardo simulados, para un caso donde no se incluyó la aparición de nuevas variantes. Tales series se construyeron contando a cada paso de la simulación el número de habitantes de la ciudad virtual que se encontraban en cada uno de los estados de salud posibles y dividiendo entre el total de agentes. Un detalle que amerita una aclaración es que en el caso de los infectados se sumaron tanto los casos asintomáticos como los sintomáticos, con el objetivo de reportar un computo global del total de enfermos.

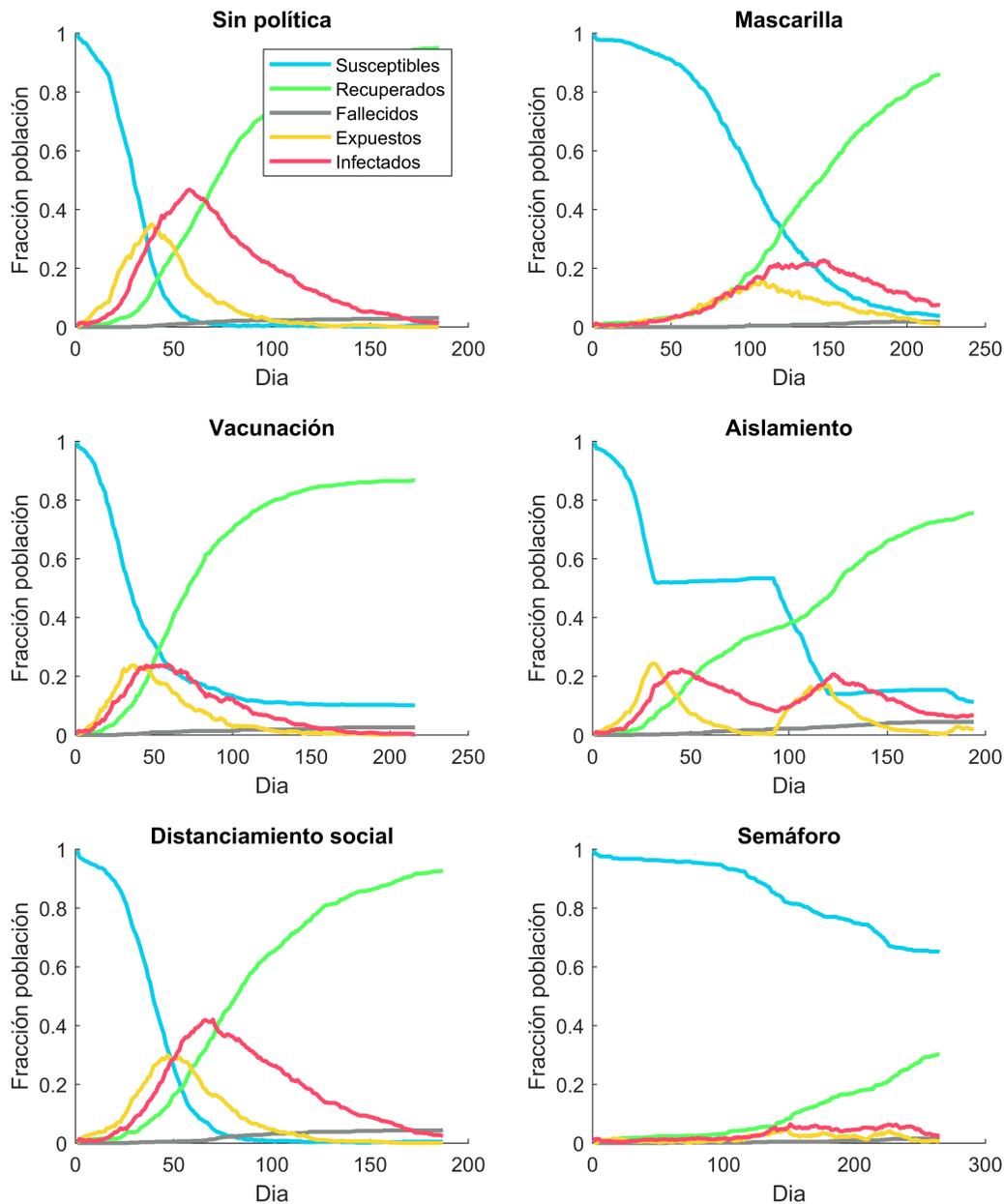


Figura 7.1: Gráficos SEIRD obtenidos en las distintas políticas de resguardo y protección implementadas.

En estos gráficos pueden apreciarse visualmente las distintas estrategias de control que ejercen las políticas implementadas sobre la pandemia. Tomando como referencia el escenario sin política, puede observarse que la mascarilla retrasa la aparición del pico

de infección y distribuye los contagios en un periodo largo de tiempo. Por otra parte la vacunación parece concentrar una cantidad menor de contagios en un lapso relativamente corto de días. Dicha estrategia de regulación es un tanto contraria a la que se consigue con medidas como el uso de cubrebocas, donde se extiende la ola de infectados y lo que se observó al hacer las pruebas fue que esta funcionaba mejor cuando las personas se vacunaban por primera vez en un periodo corto de días, por tanto, su instauración requiere de una administración adecuada. En el caso del aislamiento la curva de infección se reparte en una serie de picos pequeños, cuya altura decrece cada vez más. Dicho oleaje se genera por las sucesivas activaciones de la restricción en el movimiento de los agentes, que se dan cada vez que se supera un umbral en la ocupación hospitalaria. Respecto al distanciamiento social, puede observarse que el efecto neto de dicha política es poco notorio, obteniéndose una pequeña disminución del pico de infectados y un ligero ensanchamiento de la curva. Finalmente el semáforo, en el que se combinaron 3 de las políticas programadas mitiga casi por completo la curva de infectados. Esto implica que las medidas de control ejercen un efecto de sinergia al combinarse, lo cual en parte es producto de el comportamiento no lineal del fenómeno de estudio.

7.1.2. Series de tiempo de asistencia a distintos lugares de la ciudad

Además de los gráficos SEIRD, se exportaron con el modelo series de tiempo de la asistencia a los distintos tipos de lugar considerados dentro de la ciudad virtual, para examinar el efecto que tenían políticas como la de de aislamiento en la movilidad colectiva de los habitantes.

La figura 7.2 muestra la movilidad en distintos tipos de recinto en un caso donde no se aplicó ninguna política de resguardo, mientras que la figura 7.3 muestra los mismos gráficos en el caso donde se aplicó una política de aislamiento. Rememorando el funcionamiento de las políticas, el aislamiento se activaba toda vez que se superaba un umbral dado en la ocupación hospitalaria y se suspendían las restricciones de movimiento una vez que hubiese transcurrido un periodo fijo de 60 días. En los gráficos de asistencia a lugares dichos periodos en los que se activó el resguardo fueron indicados por un área rectangular verde.

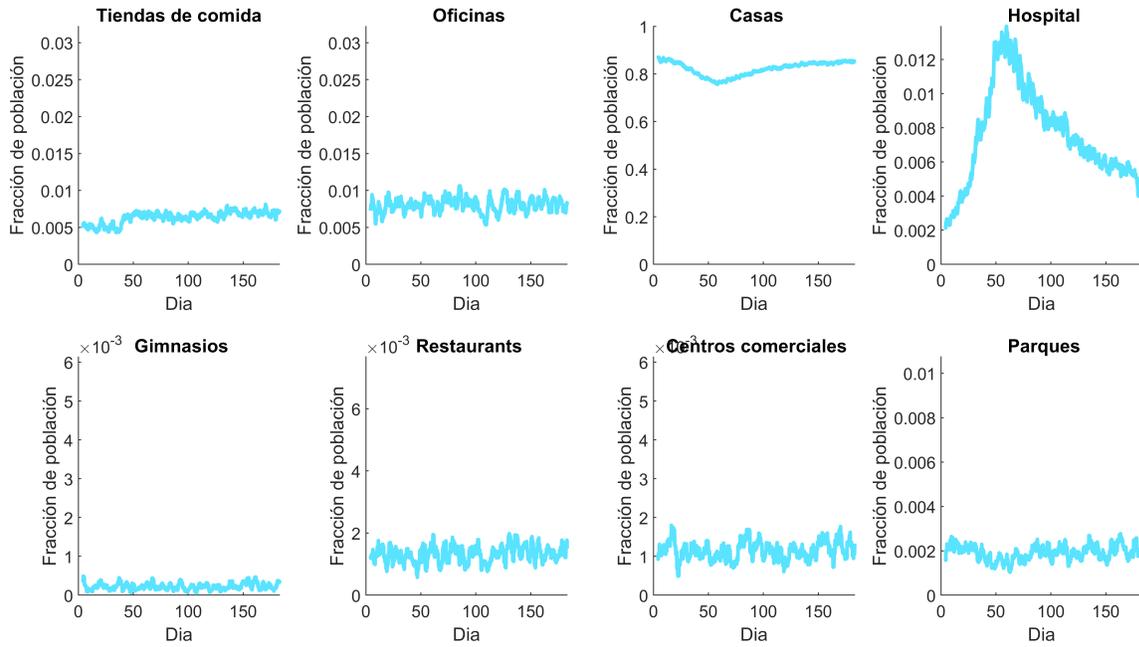


Figura 7.2: Series de tiempo de asistencia a los lugares de la ciudad simulada en el caso donde no se aplicó ninguna política de protección y resguardo.

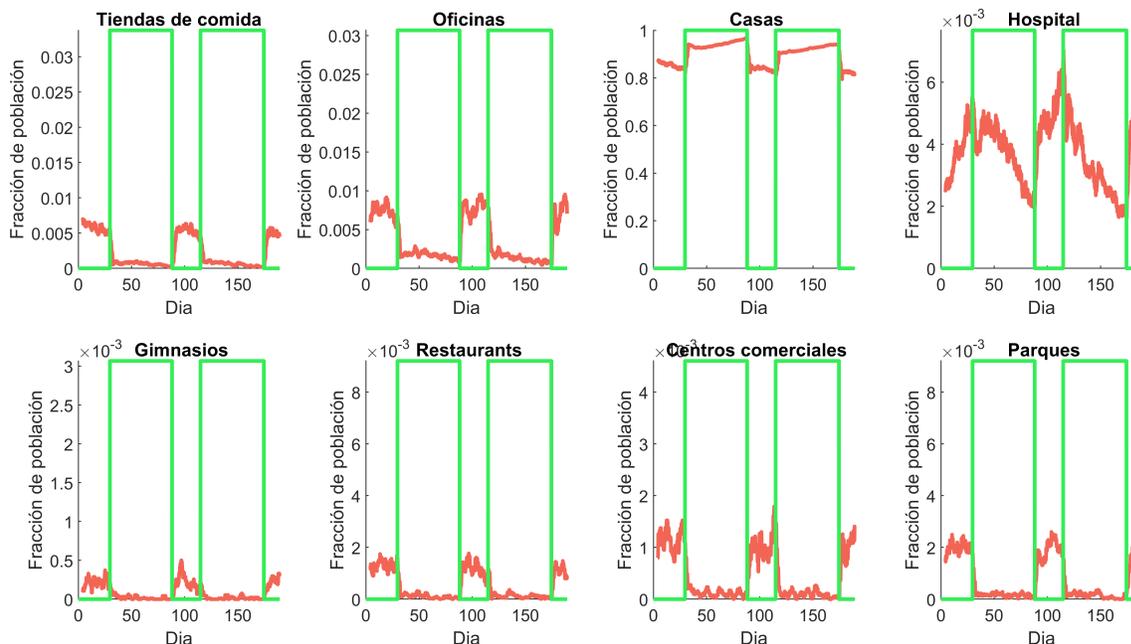


Figura 7.3: Series de tiempo de asistencia a los lugares de la ciudad simulada en el caso donde se aplicó una política de aislamiento. Los segmentos encuadrados en verde corresponden a lapsos de tiempo donde se encendieron las restricciones de movimiento debido a que se superó un umbral en la ocupación hospitalaria.

Algo que puede observarse en las series de asistencia para el caso donde no se aplicó ninguna política (Figura 7.2), es que a pesar de que no se establece ninguna restricción una fracción grande de la población permanece en casa al realizar sus actividades. Esto en realidad es producto de las condiciones en las que se aplicó la encuesta de calibración para las actividades de los habitantes virtuales. Recordando la información presentada en la sección 6.1.3, dicho cuestionario se realizó en el año 2021, en meses donde las personas habían adaptado su rutina para permanecer la mayor parte del tiempo en su hogar y así atenuar las olas de contagios. Esto resulta notorio en las distribuciones que se presentaron en la figura 6.2, ya que en todos los histogramas de tipos de lugar para realizar una actividad, la respuesta que más se eligió fue la opción "casa". Lo anterior es uno de los aspectos a mejorar de este trabajo, ya que idealmente se buscaría caracterizar el comportamiento de los habitantes cuando no hay ninguna restricción. La hipótesis que se tiene, sin embargo, es que dicho cambio resultaría en un mejor contraste entre la política nula y las demás, lo cual haría aún más notorios los resultados que se obtuvieron en esta tesis.

Observando los gráficos de asistencia correspondientes a un escenario de aislamiento (Figura 7.3), puede observarse que cuando se activan las restricciones de movimiento la población que permanece en casa aumenta, mientras que la concurrencia en otros tipos

de lugar disminuye. Respecto a la ocupación hospitalaria en particular, se aprecia que las aplicaciones sucesivas de la política tienden a liberar cada vez de mejor manera la saturación de los servicios médicos. En efecto, durante el primer periodo de cuarentena el patrón de disminución de la población que acude al hospital es más lento. La afirmación anterior puede deducirse del hecho de que la curva es cóncava hacia abajo, lo cual significa que la aceleración es pequeña en esa región (una analogía que podría resultar útil es la de una montaña rusa: la curva correspondería al inicio del trayecto de descenso, donde el carro todavía no alcanza velocidades grandes). Por el contrario, en la segunda aplicación de la política el descenso en la ocupación hospitalaria se da en forma más rápida, teniéndose una curva cóncava hacia arriba. El hecho de que el sistema responda en formas distintas al mismo estímulo nos habla de que este no es invariante en el tiempo y su comportamiento depende de la historia completa de entradas recibidas.

La idea final que se ha mencionado en el párrafo anterior puede parecer evidente a nivel teórico, ya que simular sistemas complejos implica por lo general trabajar con sistemas no lineales, que pueden contar con cierta memoria. Sin embargo, merece la pena que la establezcamos en términos más concretos de políticas: el hecho de que se aplique una medida en cierta población y esta sea efectiva, no significa que al instaurarla en otro contexto se conseguirá el mismo efecto. Incluso si las poblaciones son esencialmente idénticas, si la historia de políticas anteriores que se han probado en uno y otro contexto difieren la efectividad de la medida podría cambiar.

7.1.3. Comparación de las distintas políticas de resguardo y protección implementadas

En esta sección se hace un comparativo de diversos parámetros que pueden extraerse de las series de tiempo presentadas en las secciones anteriores, para cada una de las políticas de resguardo y protección.

Los parámetros que se calcularon para cada escenario son los que fueron esquematizados por medio de la figura 4.2 en el marco teórico. Estos son el tamaño de pico de la curva de infección h_p , el tiempo que llevó llegar a dicho pico t_p , la anchura a media altura del gráfico de infectados $FWHM$ y la fracción final de fallecidos D_∞ . Adicionalmente se obtuvo para cada política el pico de ocupación hospitalaria h_H , usando la información de asistencia a lugares exportada con el modelo. En 7.4 se muestra el valor de cada uno de los parámetros examinados para cada una de las políticas.

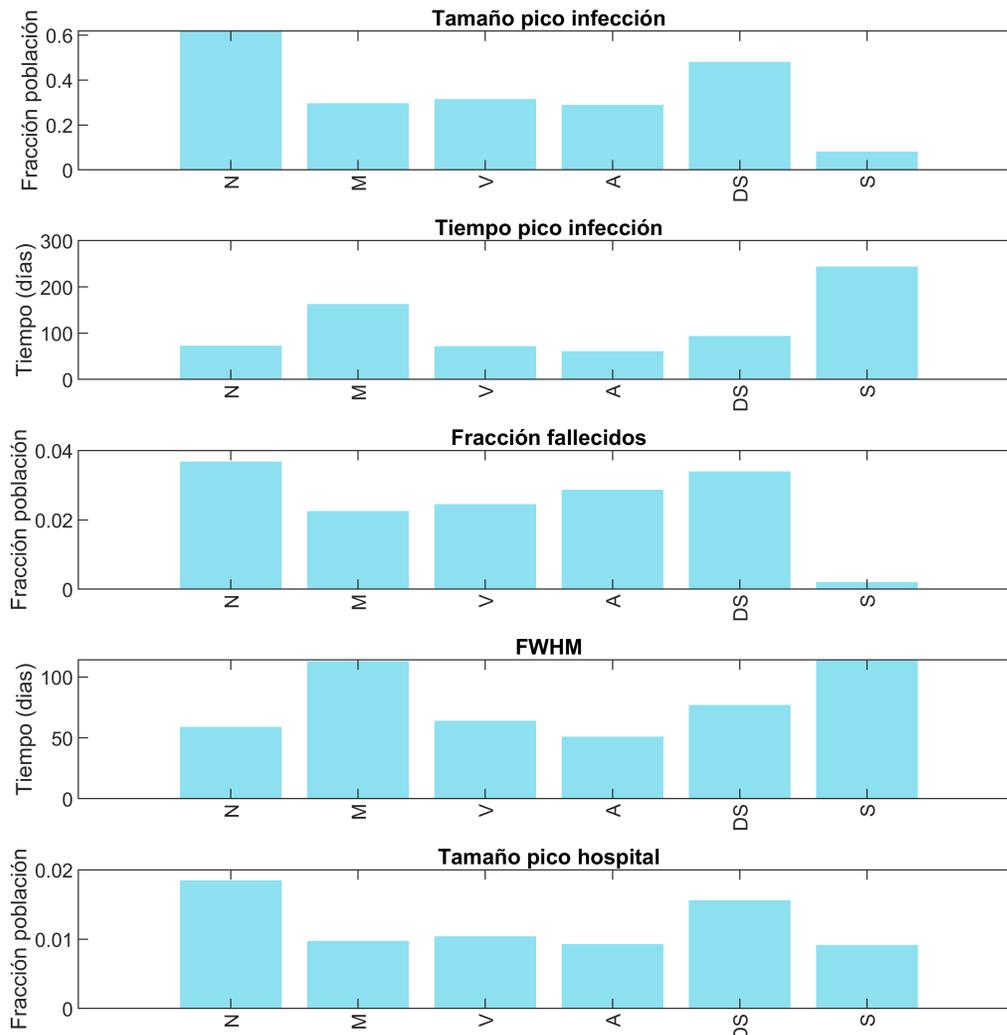


Figura 7.4: Comparación de distintas métricas obtenidas para cada uno de los escenarios de protección simulados. En la figura N = Escenario nulo (sin política), M = Uso de mascarilla, V = Vacunación, A = aislamiento, DS = Distanciamiento Social y S = Semáforo.

El propósito de la comparativa fue determinar cuál de los escenarios conseguía un mejor control de la propagación de la enfermedad. Basándonos en los parámetros que se han calculado lo que se buscaría sería un escenario en el que se consigan valores pequeños para el pico de infectados, h_p , la ocupación hospitalaria h_H y la fracción de fallecidos D_∞ y valores lo más altos posibles para el tiempo en que se llega a la cumbre de infección t_p y el ancho a media altura $FWHM$. Para combinar estos requerimientos uno podría pensar en el diseño de un parámetro nuevo, que denotaremos por e_S , el cual dependa de las variables

antes mencionadas y consiga un valor máximo en el punto donde dichas cantidades sean óptimas para atenuar la pandemia. Existen técnicas rigurosas para el diseño de este tipo de parámetros, como lo es el uso de la eficiencia de Pareto, sin embargo, dado que este era un objetivo más bien tangencial de nuestra tesis, se optó por una solución sencilla: un parámetro lineal, acotado desde 0 a 1, que combinara todas las cantidades de interés. Su definición es la que se plantea en la ecuación 7.1 y en 7.5 se muestra una figura donde se compara el valor de dicho parámetro para cada una de las políticas.

$$e_S = \left(\frac{1}{5}\right) \left((1 - h_p) + \frac{t_p}{365} + (1 - D_\infty) + (0.02 - h_H) + \frac{FWHM}{365} \right) \quad (7.1)$$

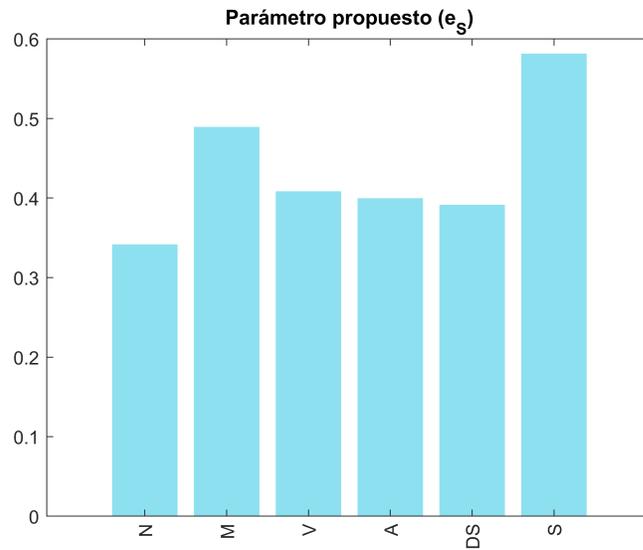


Figura 7.5: Valor del parámetro propuesto presentado en la ecuación 7.1 para distintas políticas. En la figura N = Escenario nulo (sin política), M = Uso de mascarilla, V = Vacunación, A = aislamiento, DS = Distanciamiento Social y S = Semáforo.

Puede observarse a partir de las figuras 7.4 y 7.5 que el escenario que mejor consigue un control de la pandemia es el semáforo, donde se combinaron aislamiento, distanciamiento social y uso de mascarilla. En el gráfico de barras destaca también el caso de la política de mascarilla, que por sí sola consigue extender en un periodo largo de tiempo la infección, logrando una fracción baja de fallecidos y valores pequeños para los picos de infección y ocupación hospitalaria.

Una analogía un tanto tangencial pero curiosa es que en términos generales las medidas de protección ante una pandemia siguen el mismo enfoque que se usa en las normativas de radiación, en el sentido de que se busca disminuir el tiempo de exposición (aislamien-

to), proporcionar barreras de resguardo (cubre bocas) y mantener distancia ante fuentes transmisoras (distanciamiento social). Algo que podría caracterizarse en el futuro es como depende el pico de infección de el tiempo de aislamiento de los agentes, la efectividad promedio del cubrebocas y el distanciamiento entre agentes, obteniendo funciones análogas a las exponenciales de atenuación que se tienen en radiación.

7.2. Series SEIRD al incorporar nuevas variantes.

Como se mencionó en la metodología de este trabajo, uno de los elementos novedosos que se consideraron dentro del modelo de agentes fue la aparición de nuevas variantes del virus. La comparación de escenarios presentada en las secciones anteriores de este capítulo fue realizada en condiciones donde dicha mutación del patógeno se encontraba desactivada. La razón de ello fue que al hacer las pruebas se encontró que cada escenario requería un ajuste adecuado de los parámetros usados en la ecuación 5.3, que indica la probabilidad de que un agente vuelva a ser susceptible a una nueva variante del virus después de haberse recuperado. Dado que aún es necesario definir el modo en que han de fijarse dichos parámetros acorde a la interpretación de cada política, se decidió omitir este elemento en las pruebas reportadas en este trabajo.

Para ilustrar, sin embargo, algunos ejemplos del tipo de resultados que se obtienen al considerar la mutación del virus, se muestran en la parte superior de la figura 7.6 los gráficos SEIRD para el caso donde no se aplicó ninguna política y en la parte inferior de dicha imagen las curvas correspondientes para el caso de aislamiento.

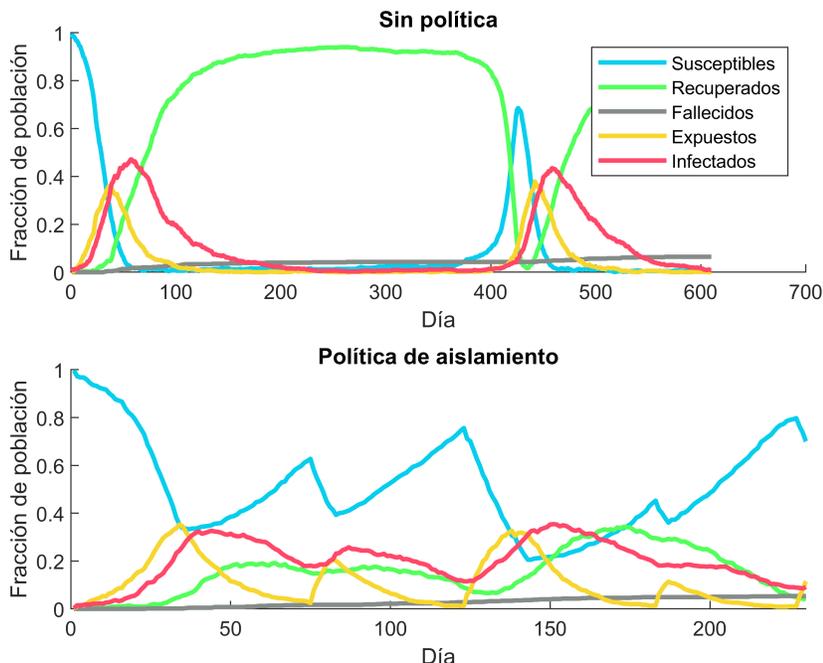


Figura 7.6: Gráficos SEIRD de los escenarios sin política y aislamiento considerando aparición de nuevas variantes del patógeno.

Los valores utilizados para la tasa de mutación y la constante de saturación en el caso sin política fueron $\mu_0 = 3$ y $C_{\text{sat}} = 10$, mientras que en el escenarios de aislamiento fue necesario considerar que dado que los agentes se aislaban la mutación se daba a un ritmo más lento. Por tal razón se modificaron los valores de dichos parámetros a $\mu_0 = 1$ y $C_{\text{sat}} = 5$.

Puede observarse que en el caso sin política se obtiene un patrón periódico en el proceso de infección, donde un solo pico se va repitiendo cada vez que aparece una nueva variante del patógeno. Por el contrario, en el escenario de aislamiento se consigue un oleaje un tanto distinto. Siguiendo la curva de infectados de izquierda a derecha se puede distinguir la aparición de un primer pico de infección, el cual se ve atenuado cuando se imponen restricciones de movimiento en la población. Una vez que se suspende la cuarentena, los habitantes vuelven a sus rutinas y se tiene como consecuencia un segundo pico de reincidencias, esta vez más pequeño. Finalmente, al aparecer otra variante se repite nuevamente dicho ciclo y puede notarse que esta vez la curva de infección se reduce mucho más como consecuencia de aplicar el aislamiento.

7.3. Clasificación de datos de movilidad

En esta sección se muestran los resultados correspondientes a la predicción del estado de salud de los agentes a partir de sus datos de movilidad.

7.3.1. Evaluación de los distintos clasificadores entrenados y comparación de la predicción al aplicar políticas de resguardo

Como se mencionó en la metodología de este trabajo, los clasificadores utilizados se entrenaron con series de posición simuladas de los agentes en dos escenarios distintos: un escenario nulo, donde los individuos no tomaban ninguna medida ante la situación de pandemia y el escenario de aislamiento, en el que se restringía su movimiento cuando la ocupación hospitalaria superaba un umbral.

Los valores obtenidos para las distintas métricas de evaluación de los clasificadores en el caso nulo (sin política) son los que se muestran en la tabla 7.1. Además de ellas se presentan en la figura 7.7 las curvas PR y ROC construidas con los modelos entrenados. Estas proveen una evaluación gráfica de la clasificación, donde los mejores clasificadores son aquellos en los que el área bajo la curva es mayor.

	SVM 1	SVM 2	RF 1	RF 2	Red neuronal	Ensamble
Accuracy	0.9206	0.9142	0.8994	0.8766	0.8390	0.9730
Precision	0.9140	0.8903	0.8789	0.8414	0.8039	0.9630
Recall	0.9294	0.9450	0.9224	0.9208	0.8903	0.9828
F1	0.9216	0.9168	0.9001	0.8793	0.8449	0.9728
F0.5	0.9170	0.9007	0.8872	0.8562	0.8198	0.9669
F2	0.9263	0.9335	0.9134	0.9038	0.8716	0.9788
ROC Score	0.9701	0.9631	0.9617	0.9496	0.9192	0.9969
PR Score	0.9661	0.9519	0.7719	0.7565	0.8838	0.9964
Briar Score	0.1514	0.1523	0.0770	0.0889	0.1139	0.0578

Tabla 7.1: Métricas de evaluación de los distintos clasificadores para un caso donde no se aplicó ninguna política de resguardo.

Como puede apreciarse tanto de la tabla de métricas, como de las curvas PR y ROC, el clasificador que mejor consigue hacer la predicción del estado de salud a partir de series de tiempo de posición es el ensamble, que combina la decisión de varios de ellos. Esto se debe a que los patrones que aprenden los distintos modelos puede ser diferente y al combinar los criterios de cada uno se consigue una decisión que pondera por igual diversos puntos de vista. Seguido al ensamble el tipo de clasificador que logró el mejor desempeño

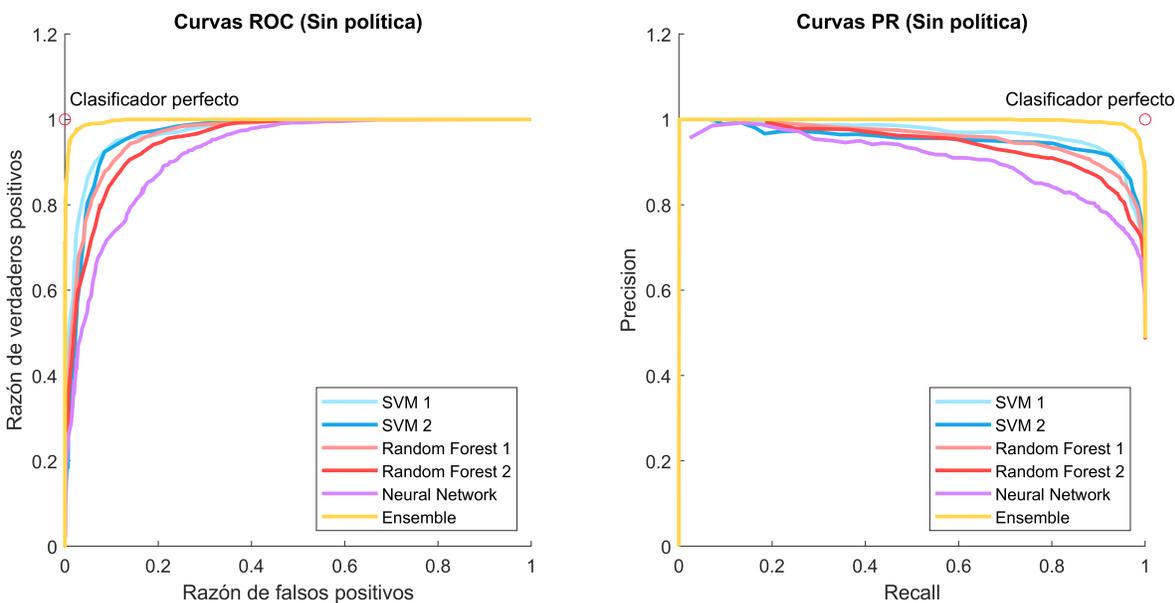


Figura 7.7: Curvas ROC y PR de los distintos clasificadores. Los datos de movilidad individual utilizados para entrenar fueron generados en un escenario sin política.

fueron los árboles aleatorios. Esto es consistente con ciertos estudios, en los cuales se ha probado que estadísticamente los árboles aleatorios y las máquinas de soporte vectorial funcionan apropiadamente para la mayoría de los problemas de clasificación que son de interés humano actualmente [30].

Otro aspecto importante a resaltar es que el modelo que obtiene los valores más bajos para las métricas evaluadas es la red neuronal. Resultaría un tanto complicado establecer a simple vista la razón de ello y es posible que modificando los parámetros (el número de neuronas intermedias sobre todo) se pudiese conseguir una mejor clasificación. Lo que podríamos afirmar es que en este caso el método de separación de los datos que usan las redes, el cual geométricamente se puede interpretar como un encasillamiento de los grupos de interés mediante planos, no es el método idóneo para clasificar los vectores de este problema. Es interesante mencionar que esta técnica suele funcionar bien en casos donde se buscan identificar estructuras complejas a partir de otras más simples, es decir problemas donde los datos pueden contar con distintos niveles de organización.

En la tabla 7.2 y la figura 7.8 se muestran respectivamente las métricas de evaluación y las curvas ROC/PR para el caso donde los clasificadores se entrenaron con datos generados en un escenario de aislamiento.

	SVM 1	SVM 2	RF 1	RF 2	Red neuronal	Ensamble
Accuracy	0.6040	0.8866	0.6711	0.7952	0.8887	0.9169
Precision	0.6455	0.8434	0.7024	0.7261	0.8400	0.8893
Recall	0.4660	0.9558	0.5686	0.9379	0.9562	0.9528
F1	0.5413	0.8961	0.6284	0.8185	0.8943	0.9200
F0.5	0.5994	0.8637	0.6708	0.7604	0.8609	0.9013
F2	0.4935	0.9310	0.5911	0.8862	0.9305	0.9394
ROC Score	0.6121	0.9407	0.7072	0.8454	0.9560	0.9759
PR Score	0.3941	0.8017	0.4220	0.7675	0.8114	0.9633
Briar Score	0.3720	0.0880	0.2951	0.1418	0.0793	0.0924

Tabla 7.2: Métricas de evaluación de los distintos clasificadores para un caso donde se aplicó una política de aislamiento.

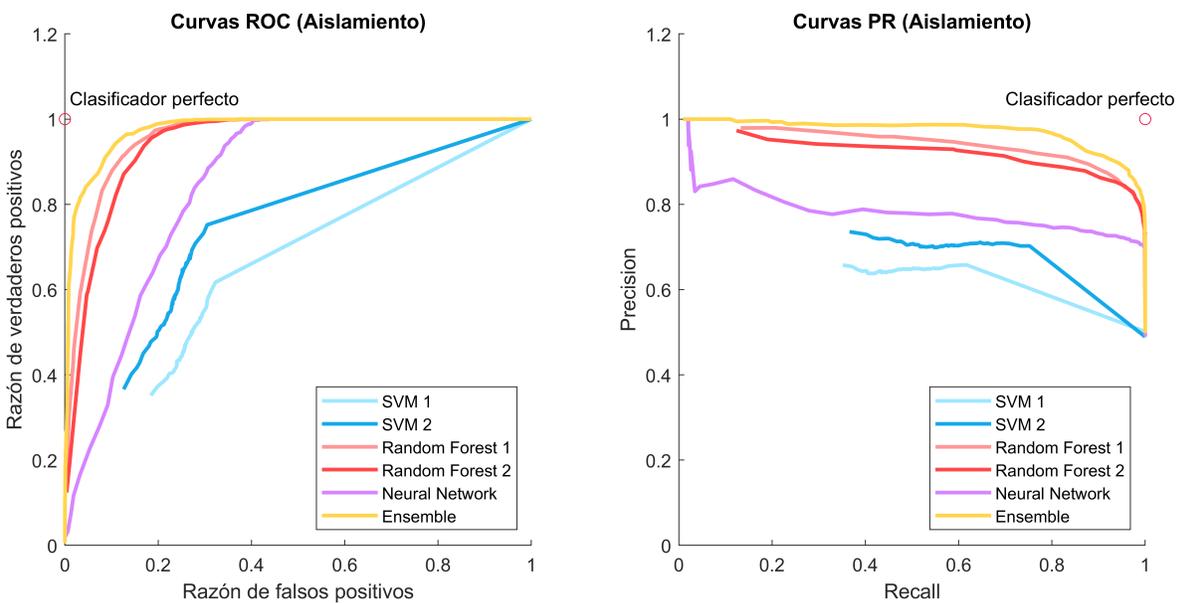


Figura 7.8: Curvas ROC y PR de los distintos clasificadores. Los datos de movilidad individual utilizados para entrenar fueron generados en un escenario de aislamiento.

En este caso nuevamente el clasificador que obtuvo el mejor resultado fue el ensamble, seguido por los árboles aleatorios que se entrenaron. Por otra parte, los modelos de clasificación que obtuvieron el peor desempeño fueron las máquinas de soporte vectorial. La razón de que en este caso dicho modelo obtenga un peor desempeño que las redes neuronales y el resto de clasificadores, podría deberse a una pérdida de continuidad en el grupo de vectores correspondientes a personas sanas. Si lo pensamos, el añadir aislamiento da lugar a dos alternativas de comportamiento para un agente sano: el comportamiento cuando no hay política y aquel que se da cuando deben resguardarse. En el espacio de características

esto podría provocar que el grupo de vectores correspondientes a personas sanas se divida en dos cúmulos separados, lo cual dificultaría el trazado de un contorno de separación entre sanos y enfermos por parte de las máquinas de soporte vectorial. Es preciso indicar que dicho modelo suele dar resultados óptimos en casos donde los datos se pueden agrupar en conjuntos convexos, i.e. conjuntos donde la línea que une a cualesquiera dos puntos también pertenece al conjunto.

Al comparar las métricas y curvas ROC/PR de los clasificadores entrenados en el caso de aislamiento con aquellos que se entrenaron en un escenario sin política, puede observarse que cuando se aplicó una medida de resguardo los modelos perdieron poder de predicción del estado de salud de las personas a partir de su movimiento. Esto podría explicarse por alguna similitud existente entre el comportamiento de los agentes sanos que se resguardan y los infectados, la cual dificulte distinguir entre miembros de dichos grupos. Una hipótesis que se tiene es que si se utilizarán en el entrenamiento datos de una cantidad más grande de personas eventualmente los clasificadores aprenderían cuales son los periodos donde se aplicó la política de aislamiento y ajustarían sus criterios de clasificación para tales lapsos de tiempo.

Algo que se puede afirmar es que el hecho de que sea más difícil realizar la predicción del estado de salud de una persona a partir de su movimiento en un escenario de aislamiento es un indicio de que dicha política está siendo efectiva, ya que refleja una disminución de la conexión entre el movimiento de los individuos y la propagación de la enfermedad. Si pensamos en que la transmisión de un virus puede ser un fenómeno multicausal impulsado por la movilidad de los individuos, diseminación del virus por el aire, etc. lo que está logrando una política de aislamiento es desacoplar hasta cierto punto la movilidad de personas de la difusión del patógeno, lo cual resulta en una disminución en la rapidez de transmisión.

Con ello se culmina la presentación y discusión de los resultados obtenidos en este trabajo. En la sección siguiente se concluirá esta tesis haciendo un contraste global de los objetivos planteados con los hallazgos realizados, mencionando también algunos de los caminos que valdría la pena explorar para mejorar y continuar con el trabajo.

Conclusión

8.0.1. Conclusiones

Uno de los objetivos planteados para este trabajo fue la elaboración de un modelo basado en agentes para simular datos de movilidad en una pandemia. Para solventarlo se diseñó e implementó en este trabajo una simulación, en la que personas van moviéndose y realizando actividades en lugares de una ciudad al mismo tiempo que se propaga una enfermedad. El modelo incluye elementos no considerados en trabajos previos que se consultaron, como son el manejo estocástico de las rutinas de actividad de los agentes, la posibilidad de calibrar el comportamiento de los mismos usando información de una encuesta y la aparición de nuevas variantes del virus dentro de la dinámica de infección. Así mismo, se implementaron varias medidas de control ante la pandemia que pueden aplicarse dentro del programa: uso de cubrebocas, aislamiento, distanciamiento social, vacunación y un semáforo que combinaba las 3 primeras.

El programa permite exportar series de tiempo de distintas variables colectivas como son las fracciones de infectados, fallecidos, la población que asiste al hospital entre otras. Comparando dichas series para las distintas políticas implementadas se llegó a la conclusión de que en el modelo el escenario más efectivo es aquel donde se combinan aislamiento, distanciamiento social y uso de cubrebocas. Dado el comportamiento no lineal del fenómeno estudiado, estas ejercen un efecto sinérgico entre sí, mitigando casi por completo el pico de infección. Así mismo, se observó que por sí solo el uso de cubrebocas provee una medida adecuada, la cual extiende en el tiempo el periodo de pandemia, atenuando el pico de infección y limitando la ocupación hospitalaria máxima y la fracción de fallecidos.

Otro objetivo importante planteado en este trabajo fue estudiar si era posible adquirir información sobre el estado de salud de una persona a partir de datos de su movimiento. Para cumplir con dicha meta se exportaron en primer lugar con el modelo de agentes series de tiempo de la posición y estado de salud de diversas personas. De dichas series se

extrajeran distintas características en los dominios de la frecuencia y wavelet, y se entrenaron clasificadores característicos del aprendizaje de máquina. Los resultados obtenidos mostraron que el modelo de clasificación que consigue las mejores métricas de evaluación es un ensamble que combina la decisión de varios de ellos, seguido de los árboles aleatorios. Así mismo, se encontró que cuando se aplicaban medidas de resguardo como el aislamiento dentro de la población los puntajes obtenidos por los modelos disminuían, lo cual implica que es más difícil realizar la predicción del estado de salud de los agentes bajo dichas restricciones de movimiento. Lo anterior indicaría una pérdida de la relación entre la movilidad de los habitantes y la transmisión del virus, lo cual pudiese tomarse como una señal de que las medidas impuestas están siendo efectivas.

Respecto a las hipótesis planteadas al principio de este trabajo podemos afirmar que la mayoría de ellas fueron confirmadas, excepto por nuestra suposición de que la vacunación obtendría uno de los mejores desempeños en la comparación de políticas. En realidad aquí es necesario observar que dicha estrategia de control sigue un enfoque distinto a las demás ya que esta no busca extender la pandemia sino acortarla mientras se disminuyen el pico de infección y la fracción de fallecidos. Por tanto el criterio utilizado, en el que se puntuaron más aquellas políticas que extendieran en el tiempo la llegada del pico de infectados, pudiese no ser idónea para evaluar la vacunación. Esto nos lleva a hacer el comentario importante de que la comparación realizada con las intervenciones sigue ciertos criterios, que pudiesen no aplicar apropiadamente para algunas estrategias de control y también podrían omitir otros aspectos importantes como son el impacto económico que puede tener la pandemia u otros factores.

8.0.2. Trabajo a futuro

Como se mencionó en la sección 7.2, uno de los detalles importantes que se encontraron al momento de realizar las pruebas de este trabajo fue que cada política requería un ajuste particular de los parámetros necesarios para simular aparición de nuevas variantes en el modelo implementado. Ante dicha dificultad se decidió en esta tesis reportar la mayor parte de resultados omitiendo esta generación de variantes. Por lo tanto el camino futuro más próximo a abordar consistirá en encontrar los valores apropiados de tales parámetros para cada uno de los escenarios y hacer una comparación de las medidas de control (cubrebocas, distanciamiento social, etc.) cuando se tienen distintas olas de infección. La razón por la que resulta de interés hacer dicho ajuste fino, es que los modelos de agentes reportados hasta ahora para simular epidemias únicamente suelen modelar un pico de infección, por lo que consideramos novedoso y pertinente publicar este modelo, donde se incorpora el

efecto cíclico que puede existir en la transmisión de un virus. Otra vía probable que se podría tomar para mejorar y continuar con el trabajo es modelar en forma más certera el movimiento de las personas dentro de la ciudad virtual. Actualmente, como se explicó en la sección 5.1.5, los agentes siguen una trayectoria lineal cada vez que se trasladan de un lugar a otro. Esto por supuesto resulta poco realista, ya que en un entorno urbano el movimiento suele restringirse a ciertas calles y vías de movilidad. Una propuesta que se tiene para considerar estas constricciones es incorporar un grafo, cuyos nodos se repartan en las calles de la ciudad. De esta forma, cada vez que uno de los agentes tuviera que trasladarse de un lugar a otro podría trazar un camino hacia su destino a través de dichos nodos. Para elegir el camino a trazar podrían primero entrenarse trayectorias usando un modelo de colonia de hormigas. Para entender como podría hacerse esto en la figura 8.1 se ha mostrado un ejemplo en el que se quieren entrenar caminos que vayan desde un punto A a un punto B.

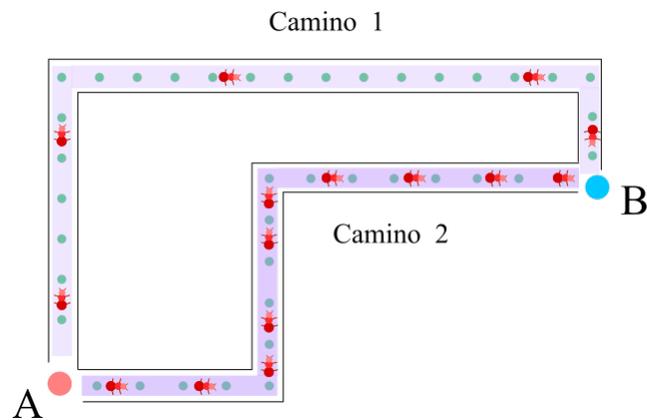


Figura 8.1: Propuesta de uso de un algoritmo de hormigas para entrenar rutas de un lugar a otro dentro de la ciudad simulada.

Lo que se hace es permitir que hormigas virtuales que comienzan su camino en A vayan explorando los distintos nodos del grafo, hasta el punto en que logren llegar a B. Una vez que una hormiga llega a la meta, se incrementa la probabilidad de que futuras hormigas tomen el camino que condujo a la hormiga hacia su destino. A dicha probabilidad dentro del contexto de este proceso se le conoce como feromona. En particular en el ejemplo de la figura hay dos caminos que pueden descubrir las hormigas. De ellos la trayectoria más corta es el camino 2 y este es el que se quedará con mayor feromona al final del proceso.

Así, lo que se haría en el modelo de agentes es usar un algoritmo de colonia de hormigas para entrenar las probabilidades de todos los caminos desde un lugar inicial a uno final. Una vez hecho esto, cuando un agente deba transportarse entre dichos lugares se seleccionaría alguno de los caminos usando las probabilidades entrenadas. Es importante notar que bajo

tal esquema los caminos mas cortos son los de mayor probabilidad y los de más duración serán los que menos frecuentarán los agentes. Esto en realidad es consistente con el criterio de selección de rutas que suelen hacer los habitantes de una ciudad, ya que por lo general se procura tomar la vía más rápida, aunque existe la posibilidad de que se deban recorrer otros caminos en ocasiones.

Bibliografía

- [1] Muhammad Shahroz et al. Covid - 19 digital contac tracing applications and techniques: A review post initial deployments. *Transportation Engineering*, September 2021.
- [2] Leonie Kahnbach et al. Quality and adoption of covid - 19 tracing apps and recommendations for development: Systematic interdisciplinary review of european apps. *Journal of Medical Internet Research*, 23, June 2021.
- [3] Facebook. Our work on covid - 19. <https://dataforgood.fb.com/docs/covid19/>.
- [4] Apple. Informes de tendencias de movilidad. <https://covid19.apple.com/mobility>.
- [5] Google. Informes de movilidad local sobre el covid-19. <https://www.google.com/covid19/mobility/>.
- [6] Pierre Nouvellet et al. Reduction in mobility and covid - 19 transmission. *Nature Communications*, 12, February 2021.
- [7] Qianqian Cui, Zhipeng Qiu, Wenbin Liu, and Zengyun Hu. Complex dynamics of an sir epidemic model with nonlinear saturate incidence and recovery rate. *entropy*, June 2017.
- [8] Stuart Rusell and Peter Norvig. *Artificial Intelligence a modern approach*. Prentice hall, Englewood Cliffs, New Jersey 07632, 1995.
- [9] Petrônio C.L. Silva, Paulo V.C. Batista, Hélder S. Lima, Marcos A. Alves, Frederico G. Guimarães, and Rodrigo C.P. Silva. Covid-abs: An agent-based model of covid-19 epidemic to simulate health and economic effects of social distancing interventions. *Chaos, Solitons & Fractals*, 139:110088, 10 2020.

- [10] Jonatan Gomez, Jeisson Prieto, Elizabeth Leon, and Arles Rodríguez. Infekta—an agent-based model for transmission of infectious diseases: The covid-19 case in bogotá, colombia. *PLOS ONE*, 16:e0245787, 2 2021.
- [11] Adam Catching, Sara Capponi, Ming Te Yeh, Simone Bianco, and Raul Andino. Examining the interplay between face mask usage, asymptomatic transmission, and social distancing on the spread of covid-19. *Scientific Reports*, 11:15998, 12 2021.
- [12] Vittoria Offeddu, Chee Fu Yung, Mabel Sheau Fong Low, and Clarence C Tam. Effectiveness of masks and respirators against respiratory infections in healthcare workers: A saystematic review and meta - analysis. *Clinical Infectious Diseases*, 65:1934 – 1942, December 2017.
- [13] Ruslan I. Mukhamadiarov, Shengfeng Deng, Shannon R. Serrao, Priyanka, Riya Nandi, Louie Hong Yao, and Uwe C. Täuber. Social distancing and epidemic resurgence in agent-based susceptible-infectious-recovered models. *Scientific Reports*, 11:130, 12 2021.
- [14] Antonio A. Aguilera, Ramon F. Brena, Oscar Mayora, Erik Molino-Minero Re, and Luis A. Trejo. Multi-sensor fusion for activity recognition: A survey. *Sensors*, 19, August 2019.
- [15] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits on predictability in human mobility. *Science*, 327, February 2010.
- [16] Alan v. Oppenheim and Alan S. Willsky. *Signals and Systems*. Prentice Hall Signal Processing Series, Upper Saddle Riv er, New Jersey 07458, 1997.
- [17] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press An Imprint of Elsevier, 525 B Street, Suite 1900, San Diego, California 92101 - 4495, USA, 1998.
- [18] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 3600 Univesity City Science Center, Philadelphia Pennsylvania 19104-2688, 1992.
- [19] Juan Uriel Legaria Peña. Wavelet transform. https://github.com/walup/Signal_Analysis/tree/main/WaveletTransform.
- [20] Juan Uriel Legaria Peña. Pandemic activities. <https://github.com/walup/Pandemic-Activities>.

- [21] Ken Perlin. Making noise. <https://web.archive.org/web/20160308022101/http://noisemachine.com/talk1/index.html>, December 1999.
- [22] Christina Maria Mayr and Gerta Köster. Social distancing with the optimal steps model. *Collective Dynamics*, December 2021.
- [23] Amit Agrawal and Rajneesh Bhardwaj. Probability of covid-19 infection by cough of a normal person and a super-spreader. *Physics of Fluids*, March 2021.
- [24] Centers for Disease Control and Prevention. Covid - 19 symptoms. <https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html>.
- [25] Sanjay Mishra. How long does covid-19 linger in your body? new report offers clues. <https://www.nationalgeographic.com/science/article/persisting-coronavirus-could-drag-out-covid-19-symptoms>.
- [26] M.P.H. Lisa Maragakis, M.D. How long does covid-19 linger in your body? new report offers clues. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/diagnosed-with-covid-19-what-to-expect>.
- [27] Qiuyue Ma, Jue Liu, Qiao Liu, et al. Global percentage of asymptomatic sars-cov-2 infections among the tested population and individuals with confirmed covid-19 diagnosis. *JAMA Network Open*, 4, December 2021.
- [28] Brenda Goodman. Infected, vaccinated or both: How protected am i from covid? <https://www.webmd.com/lung/news/20211110/how-covid-immunity-works>.
- [29] Johns Hopkins Coronavirus Resource Center. Mortality analyses. <https://coronavirus.jhu.edu/data/mortality>.
- [30] Manuel Fernández Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, October 2014.
- [31] Emanuele Galli, Leticia Cuéllar, Stephan Eidenbenz, Mary Ewers, Sue Minszewski, and Christof Teuscher. Activitysim: Large-scale agent-based activity generation for

infrastructure simulation. *Proceedings of the 2009 Spring Simulation Multiconference*, 2009.