



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Un entorno de agentes para simular la expansión de un virus basado en el comportamiento social y las medidas de contención

Estudiante: Manuel Seijas Carpentel
Directoras: Bertha Guijarro Berdiñas
Amparo Alonso Betanzos
Alejandro Rodríguez Arias

A Coruña, febreiro de 2021.

a todos mis compañeros de vida

Agradecimientos

A mis padres y hermanos, por estar ahí durante todo el proceso. A Veá, por aguantarme todo los días y noches que supuso el trabajo, que no fueron pocas. Quiero darle las gracias también a mis compañeros de clase, y a Bieito y Marco, por estar ahí y apoyarme estos últimos cuatro años, creo que sin ellos no hubiese sido posible llegar hasta aquí.

Resumen

La crisis del coronavirus COVID-19 ha generado importantes cambios en la sociedad, obligando a los distintos países a adoptar una serie de medidas de control que, aun compartiendo algún fondo común, han sido diferentes en cada uno de ellos, con diferentes resultados y respuestas en sus poblaciones e individuos. Las respuestas a cuestiones del tipo ¿Cómo actuarían las personas frente al virus? ¿Cómo se modificaría el comportamiento de estas personas dependiendo de las medidas de contención del virus? ¿Cómo variaría la expansión del virus, así como sus tasas de afectación y características dependiendo de la respuesta humana? son relevantes a la hora de actuar frente a una expansión viral de estas características. Disponer de esta información no solo puede ayudar a prever el comportamiento del virus y paliarlo de una forma más sencilla, sino que puede ayudar a encontrar aquellas medidas que optimicen a la vez el control de la expansión y la salud mental y satisfacción personal de los individuos.

Un agente software es un ente que contiene información, atributos y exhibe un comportamiento. La simulación basada en agentes es una técnica computacional que permite modelar procesos como sistemas dinámicos de agentes que interactúan entre sí. En este trabajo, se implementa un modelo basado en agentes que representarán personas en una sociedad. Cada uno de estos agentes tendrá un comportamiento específico en función de ciertas necesidades sociales, intenciones y creencias que resultan determinantes a la hora del cumplimiento de una normativa, como puede ser el confinamiento o la distancia social. Además, estos agentes influyen en y se ven influenciados por el comportamiento y opinión de sus redes de contactos (familiares, amigos, compañeros de trabajo, gente relevante...) de modo que un agente puede variar su propio comportamiento a lo largo del tiempo.

Este entorno, por tanto, nos permite modelar una simulación de una propagación epidemiológica y analizar cómo el comportamiento, la aceptación o no de las normas, y la variación de las características de los agentes influyen en la propagación. Para poder adquirir datos sobre los agentes se usarán, también, cuestionarios desarrollados por una especialista en psicología, que han sido difundidos en internet.

En definitiva, el modelo permitirá simular y evaluar distintas medidas políticas y sociales, que pueden ir desde el cierre de fronteras o la vigilancia ciudadana hasta el confinamiento, para tratar de encontrar el mejor equilibrio entre la satisfacción y aceptación de las normas por parte de los ciudadanos y su eficacia.

Abstract

The COVID-19 coronavirus crisis has generated important changes in society, forcing the different countries to adopt a series of control measures that, even sharing some common fund, have been different in each of them, with different results and responses in their populations and individuals. The answers to questions such as: How would people act against the virus? How would the behavior of these people change depending on the measures to contain the virus? How would the spread of the virus vary, as well as its rates of affectation and characteristics depending on the human response? They are relevant when it comes to dealing with a viral expansion of these characteristics. Having this information can not only help to anticipate the behavior of the virus and alleviate it in a simpler way, but it can also help to find those measures that optimize the control of expansion and the mental health and personal satisfaction of individuals.

A software agent is an entity that contains information, attributes, and exhibits behavior. Agent-based simulation is a computational technique that allows processes to be modeled as dynamic systems of agents that interact with each other. In this work, a model based on agents that will represent people in a society is implemented. Each of these agents will have a specific behavior based on certain social needs, intentions and beliefs that are decisive when it comes to compliance with regulations, such as confinement or social distance. In addition, these agents influence and are influenced by the behavior and opinion of their contact networks (family, friends, co-workers, relevant people ...) so that an agent can vary its own behavior over time. The environment, therefore, allows us to model a simulation of an epidemiological spread and analyze how behavior, the acceptance or not of the norms, and the variation in the characteristics of the agents influence the spread. In order to acquire data on the agents, questionnaires developed by a specialist in psychology, which have been disseminated on the internet, will also be used. In short, the model will allow simulating and evaluating different political and social measures, which can range from the closure of borders or citizen surveillance until confinement, to try to find the best balance between the satisfaction and acceptance of the rules by citizens and their effectiveness.

Palabras clave:

- Agentes software inteligentes
- Modelado basado en agentes
- Inteligencia Artificial
- NetLogo

- COVID-19
- Modelo epidemiológico

Keywords:

- Intelligent software agents
- Agent-Based modeling

-
- Artificial Intelligence
 - NetLogo

- COVID-19
- Epidemiological model

Índice general

1	Introducción	1
1.1	Contextualización del problema	2
1.2	Estructura de la memoria	3
2	Planificación y costes	5
2.0.1	Planificación temporal	5
2.0.2	Costes	8
3	Agentes Inteligentes	9
3.1	Agentes inteligentes	9
3.2	Modelado basado en agentes	10
4	HUMAT	13
4.1	Evaluación de las alternativas de comportamiento	14
4.2	Selección de la alternativa preferida	15
4.3	Actuación y efectos experienciales	15
4.4	Actualización de la memoria y valores	16
5	Modelos epidemiológicos	17
5.1	Modelos epidemiológicos	17
5.2	Modelo SIR	18
6	Herramientas utilizadas	21
6.1	NetLogo	21
6.2	Microsoft Teams	23
6.3	Git	23
7	Desarrollo del proyecto	25
7.1	Método de trabajo	25

7.2	Sprint 1: Modelo Epidemiológico Simple	25
7.2.1	Escenario inicial	26
7.2.2	Implementación del modelo SI	27
7.2.3	Estudio de los resultados	29
7.3	Sprint 2: Modelo SIR Completo	30
7.3.1	Definición del modelo SIR completo	30
7.3.2	Implementación del modelo	30
7.3.3	Análisis de resultados	33
7.4	Sprint 3: Incorporación de las conexiones	36
7.4.1	Implementación de la red	36
7.5	Sprint 4: Obtención y análisis de datos	37
7.5.1	Obtención de datos mediante encuesta	37
7.5.2	Limpieza de los datos	37
7.6	Sprint 5: HUMAT: Definición base y selección de alternativas	38
7.6.1	Definición e implementación de las variables base	38
7.6.2	Evaluación y selección de alternativas de comportamiento	39
7.7	Sprint 6: Revisión del modelo epidemiológico	40
7.7.1	Ampliación del modelo SIR	40
7.7.2	Uso de la red de conexiones	41
7.7.3	Implementación de las medidas de contención	41
7.7.4	Ajustes menores	43
7.8	Sprint 7: Mejora de la red de conexiones y salida del sistema	44
7.8.1	Red social	44
7.9	Sprint 8: Completando la arquitectura HUMAT	45
7.9.1	Añadiendo un nuevo grupo de necesidades	45
7.9.2	Implementación de las disonancias	46
7.9.3	Mejora en la selección de alternativas	47
7.9.4	Preguntas y señales	49
7.10	Sprint 9: Añadiendo los datos	51
7.10.1	Organización de los datos	51
7.10.2	Tratamiento e importación de los datos	53
8	Resumen de funcionalidades y características	57
9	Pruebas y resultados	61
9.1	Primera prueba	61
9.2	Segunda prueba	62
9.3	Tercera prueba	65

10 Conclusiones y trabajo futuro	67
Bibliografía	69

Índice de figuras

2.1	Primeros dos trimestres de la planificación	6
2.2	Últimos dos trimestres de la planificación	7
6.1	Biblioteca de modelos de Netlogo	22
6.2	Ejemplo de interfaz para el modelo epiDEM Basic	22
6.3	Logo de Microsoft Teams	23
6.4	Logo de Git	23
7.1	Escenario inicial	26
7.2	Estado de la interfaz	28
7.3	Simulación en la Fase 1 del desarrollo	29
7.4	Resultados Fase 1	31
7.5	Transiciones Fase 2	31
7.6	Interfaz Fase 2	32
7.7	Simulación 1 en la Fase 2 del desarrollo	34
7.8	Simulación 2 en la Fase 2 del desarrollo	35
7.9	Red inicial	42
7.10	Diagrama de estados del modelo SIR ampliado	42
7.11	Visualización de las conexiones a mitad del proceso de expansión del virus	45
9.1	Configuración para la primera prueba	62
9.2	Resultados para la primera prueba con medida de contención nula. En verde la línea de individuos susceptibles a lo largo de la simulación, en naranja el número de infectados, en rojo el número de muertos y en gris el número de inmunes.	63
9.3	Prueba 2: Final de la simulación. En verde la línea de individuos susceptibles a lo largo de la simulación, en naranja el número de infectados, en rojo el número de muertos y en gris el número de inmunes.	64

9.4 Prueba 3: Final de la simulación. En verde la línea de individuos susceptibles a lo largo de la simulación, en naranja el número de infectados, en rojo el número de muertos y en gris el número de inmunes. 66

Índice de tablas

7.1	Tabla de disonancias	48
-----	--------------------------------	----

Introducción

LA crisis del coronavirus COVID-19 ha generado importantes cambios en la sociedad, obligando a los distintos países a adoptar una serie de medidas de control que, aun compartiendo algún fondo común, han sido diferentes en cada uno de ellos, con diferentes resultados y respuestas en sus poblaciones e individuos. Las respuestas a cuestiones del tipo ¿Cómo actuarían las personas frente al virus? ¿Cómo se modificaría el comportamiento de estas personas dependiendo de las medidas de contención del virus? ¿Cómo variaría la expansión del virus, así como sus tasas de afectación y características dependiendo de la respuesta humana? son relevantes a la hora de actuar frente a una expansión viral de estas características. Disponer de esta información no solo puede ayudar a prever el comportamiento del virus y paliarlo de una forma más sencilla, sino que puede ayudar a encontrar aquellas medidas que optimicen a la vez el control de la expansión y la salud mental y satisfacción personal de los individuos.

Un agente software es un ente que contiene información, atributos y exhibe un comportamiento. La simulación basada en agentes es una técnica computacional que permite modelar procesos como sistemas dinámicos de agentes que interactúan entre sí. En este trabajo, se implementa un modelo basado en agentes que representarán personas en una sociedad. Cada uno de estos agentes tendrá un comportamiento específico en función de ciertas necesidades sociales, intenciones y creencias que resultan determinantes a la hora del cumplimiento de una normativa, como puede ser el confinamiento o la distancia social. Además, estos agentes influyen en y se ven influenciados por el comportamiento y opinión de sus redes de contactos (familiares, amigos, compañeros de trabajo, gente relevante...) de modo que un agente puede variar su propio comportamiento a lo largo del tiempo.

La arquitectura utilizada para los agentes será la HUMAT [1], que define diferentes características de comportamiento para los agentes que formarán parte de una sociedad, así como las comunicaciones entre los mismos, dando lugar a una red social con capacidades de evolución en sus decisiones y características. Además, incluye un método de selección de

alternativas de comportamiento, lo cual funciona adecuadamente con la idea de aplicar diferentes medidas de contención dentro del modelo, así como dar libertad a los agentes para seguirlas o no.

Este entorno, por tanto, nos permite modelar una simulación de una propagación epidemiológica y analizar cómo el comportamiento, la aceptación o no de las normas, y la variación de las características de los agentes influyen en la propagación. Para poder adquirir datos sobre los agentes se usarán cuestionarios desarrollados por una especialista en psicología, que ya fueron difundidos en internet, entre marzo y abril de 2020.

En definitiva, el modelo permitirá simular y evaluar distintas medidas políticas y sociales, que pueden ir desde el cierre de fronteras o la vigilancia ciudadana hasta el confinamiento, para tratar de encontrar el mejor equilibrio entre la satisfacción y aceptación de las normas por parte de los ciudadanos y su eficacia.

1.1 Contextualización del problema

El Coronavirus SARS-COV-2, en la actualidad, es el mayor foco de preocupación en el mundo, innumerables estudios y trabajos se han basado en la investigación sobre este virus epidemiológico.

Pues bien, como ya se ha comentado, los gobiernos de los diferentes países han tenido que frenar la expansión del virus mediante la toma de diferentes medidas de contención, lo que supone un control determinado del virus (con sus valores propios y características), así como de los determinados niveles de satisfacción y control por parte de la población. En esta situación, es importante poder simular la expansión del virus en una sociedad dadas determinadas características, y es que no solo es importante para conocer como se expande el propio virus, sino también para comprobar cómo la influencia de la población afecta a los valores y tasas de expansión del virus. Por ello se propone el proyecto, para lograr llegar a estas simulaciones y, gracias a ellas, encontrar las mejores soluciones y actuaciones frente a este u otro virus de similares características.

El entorno planteado simulará una sociedad en la que existirán diferentes elementos, como personas, relaciones entre estas, etc. Teniendo esto en cuenta, se da una situación perfecta para la representación del problema mediante un modelo basado en agentes, por diferentes motivos:

- La representación gráfica es perfecta para la situación, los agentes serán fácilmente representables, así como las relaciones entre ellos y sus características. Además, la expansión del virus y las tasas de infectados/sanos/recuperados serán visibles de una manera

sencilla.

- Los agentes tendrán repercusión directa sobre la expansión del virus, ya que aceptar o no las medidas de contención permitirá una mayor o menor expansión del virus
- Las relaciones entre agentes podrán variar y ser fácilmente representables y controladas.

1.2 Estructura de la memoria

A continuación, se muestra la estructura de los capítulos siguientes, explicando brevemente su contenido.

En el capítulo 2, de **planificación y costes**, se marcarán aquellos detalles necesarios en cuanto a la planificación del proyecto, duración, organización, costes, etc.

A continuación se hablará de los **agentes inteligentes**. Este capítulo pretende dejar claros todos los conceptos acerca de los mismos y del modelado basado en agentes para que tengan sentido todas las explicaciones que se realizan durante el capítulo de desarrollo.

Tras hablar de los agentes inteligentes, es importante concretar información acerca de la arquitectura **HUMAT**, que definirá el comportamiento de los entes que componen el sistema. En este capítulo se hablará de todas las etapas del "ciclo de vida" del agente, que después se implementarán en el proyecto.

El siguiente punto importante es hablar acerca de los **modelos epidemiológicos**, que, como comentaba anteriormente, definirán el comportamiento del sistema (que no de los agentes).

Por otro lado, también es importante hablar de las **herramientas utilizadas**, en concreto Netlogo, Teams y Git, que supusieron una gran ayuda para el desarrollo del proyecto.

Tras explicar todos los conceptos anteriores, se entra en materia con el **desarrollo del proyecto**, un capítulo dedicado a todos los Sprints que formaron el proyecto, su modo de ejecución y cómo se han planteado. Este capítulo es el más extenso, ya que abarca todos los meses de trabajo sobre el proyecto.

Tras hablar sobre el desarrollo del proyecto, y por último, es importante acabar con unas **conclusiones y trabajo futuro**, donde se realiza un pequeño resumen final del proyecto y de sus resultados, y se marcarán unas pautas a seguir para poder continuar con el mismo, y mejorarlo.

Planificación y costes

EL modelo de trabajo seleccionado fue el de Agile en Sprints. Podemos definir que la duración de un Sprint es de un mes, con reuniones semanales para ver la ejecución del mismo y como avanza. Pues bien, la planificación es de un total de 10 meses, lo que equivale a 10 sprints. Toda la planificación puede verse en las figuras 2.1 y 2.2, mediante un diagrama de Gantt que determina la duración y la organización del proyecto. Como puede verse en las mismas, se divide la planificación en cuatro trimestres, que suponen diferencias de costes temporales en la organización.

2.0.1 Planificación temporal

En el primer trimestre la carga de trabajo es menor en el desarrollo, no porque la implementación de lo planteado sea difícil, sino porque supone aprender a utilizar las herramientas (en concreto, el lenguaje de NetLogo) y definir los conceptos que después se aplicarán durante el proyecto. Por eso, a pesar de que las tareas sean "rápidas" de ejecutar y deberían de suponer un coste menor, igualmente se le asigna el coste mensual, para paliar la curva de aprendizaje.

El segundo trimestre es un punto grande de implementación. Ya se ha cubierto la curva de aprendizaje y se puede comenzar a implementar la base de la arquitectura HUMAT, así como mejorar las implementaciones anteriores para conseguir un código y definiciones más limpios.

El tercer trimestre sigue el mismo camino, salvo que, además, se dedica un tiempo a aplicar los datos al sistema, lo que empieza a dar resultados.

Por último tenemos el cuarto trimestre, en el que el desarrollo se concentra en el primer mes, enero, que se dedica a realizar pruebas y análisis de los resultados del sistema. Sin embargo, se ha decidido añadir aquí también la escritura de la memoria y la presentación del proyecto, que suponen gran parte de los meses de febrero y marzo, y que se estima añadirlo como coste al desarrollo.



(a) Primer trimestre



(b) Segundo trimestre

Figura 2.1: Primeros dos trimestres de la planificación



(a) Tercer trimestre



(b) Cuarto trimestre

Figura 2.2: Últimos dos trimestres de la planificación

2.0.2 Costes

Como se ha indicado en la planificación temporal, la duración del proyecto es de 10 meses. La dedicación al proyecto no fue completa, debido al horario de trabajo, lo que supuso unas 3 horas diarias de trabajo sobre el mismo. Esto, entonces, recae en 600 horas (3 horas diarias y 5 días a la semana, suponiendo descanso el fin de semana). Suponiendo que el sueldo supondría el del convenio colectivo 22.993,74€, lo que suponen 11.98€ (aproximadamente) cada hora. Pues bien, esto arrojaría un coste personal del proyecto de 7188€. No se añaden costes de software ni hardware, por lo que ese sería el coste final del mismo.

Agentes Inteligentes

Para comenzar a hablar de un sistema basado en agentes, es importante definir de manera correcta lo que es un Agente Inteligente: un sistema informático, o entidad, que es capaz de percibir su entorno, procesar esas percepciones y actuar en respuesta a las mismas. El objetivo del agente es maximizar una medida deseada, que puede ser, por ejemplo, alcanzar una meta en un número menor de movimientos.

3.1 Agentes inteligentes

Según Nikola Kasabov [2] existen diferentes características que los Agentes Inteligentes deben poseer para considerarse como tales:

- Crear nuevas **reglas de resolución** de un problema de forma incremental, esto es, que sea capaz de generar sus propias reglas de actuación frente a los problemas o situaciones de decisión planteadas.
- Capacidad de **adaptarse** en tiempo de ejecución.
- Capacidad de **valorarse** a sí mismos en términos de comportamiento, errores y éxitos.
- **Aprenden y mejoran** por medio de la interacción con el entorno.
- Aprenden más rápido a mayor cantidad de datos accesibles (mayor acierto en la simulación del agente software).
- Pueden tener una cierta **capacidad memorística** basada en ejemplos.
- Contienen parámetros, capaces de determinar características de corto y largo plazo, como pueden ser: edad, capacidad de recuerdo, necesidades sociales, etc.

- Tienen la habilidad de comunicarse (habilidad social). No tienen un protocolo de comunicación como tal, pero sí que pueden transmitirse datos y tomar decisiones de acuerdo a ellos.

Como podemos ver, varias de estas características proponen que el agente esté posicionado en un **mundo o escenario**, ya sea virtual o real, y es que esta es, junto con la capacidad del agente de ser **autónomo**, la característica más importante de los mismos.

Estos escenarios, a su vez, poseen diferentes propiedades que determinarán, en cierta medida, la actuación del agente:

- **Dinámicos o estáticos:** el agente tendrá conciencia de si el escenario cambiará mientras él actúe o tome decisiones.
- **Accesible o innacesible:** el agente tendrá mayor o menor información sobre el escenario.
- **Determinista o no-determinista:** si al realizar una acción sobre el mismo, la consecuente reacción está asegurada.
- **Discreto o continuo:** si se limita el número de acciones y percepciones que se pueden dar dentro del escenario.

A pesar de que los agentes pueden basarse en estas características para considerarse como tal, según Weiss [3] existen diferentes arquitecturas posibles para los agentes:

- **De base lógica:** Son aquellos donde la decisión de actuación frente a un problema o decisión se toma bajo una base de deducción lógica.
- **Reactivos:** La decisión se toma mediante una relación directa entre situación y acción.
- **Creencia-deseo-intención:** Agentes en los que la toma de decisión depende de la manipulación de datos relacionados con los valores, deseos e intenciones del agente.
- **Arquitectura por capas:** Agentes creados con diferentes capas, cada cual con más o menos razonamiento frente al mismo problema.

3.2 Modelado basado en agentes

Los modelos basados en agentes (MBA) [4] constituyen una nueva generación de métodos computacionales que permiten modelar la estructura de un sistema complejo y simular su evolución dinámica a lo largo del tiempo. Tienen un amplio uso en las ciencias sociales de

la época moderna, sin embargo, su campo de aplicación no se reduce a esto, aunque su uso todavía no esté muy extendido.

Un modelo basado en agentes es un tipo de modelo computacional basado en identificar el comportamiento individual de las entidades (personas, animales, cosas, etc.) que componen un sistema, y se usa para analizar qué cambios introducir en el sistema real para lograr el comportamiento deseado.

Como vemos, por propia definición, se aplica perfectamente al caso planteado. Podremos identificar el comportamiento individual de las personas que forman nuestra población simulada, y ver como implementar las medidas de contención en un entorno real gracias a las simulaciones en el ficticio.

Los MBA pueden combinar elementos de teoría de juegos, sistemas complejos, emergencia, sociología computacional y programación evolutiva, aunque no debe confundirse con ninguno de ellos. Como se ha comentado, los modelos simulan las operaciones simultáneas de los agentes en un intento de recrear y predecir las acciones de fenómenos complejos.

Generalmente, los agentes individuales actúan según lo que perciben como intereses propios, aunque variará en nuestro caso gracias a la red social.

Según Ariel Quezada y Enrique Canessa [5], el uso de este tipo de modelos tiene diferentes beneficios, a destacar:

- Es fácil modelar la emergencia de fenómenos "bottom up", en los que la interacción de los agentes propicia comportamientos cualitativamente distintos a nivel global. En el caso estudiado, esto se ve en las diferentes alternativas de comportamiento que puede tomar el agente, así como la adherencia a las diferentes medidas de contención del virus.
- Se pueden modelar agentes con comportamientos homogéneos.
- Los agentes tienen racionalidad y conocimiento limitado, lo que permite modelar individuos mucho más "reales".
- Se pueden realizar experimentos controlados, lo que es mucho más difícil en un entorno real.
- Se puede cambiar fácilmente tanto el nivel de análisis como el tamaño de muestra.

Sin embargo, también tienen algunas limitaciones:

- El modelo es difícil de validar.

- Igual que en muchas aplicaciones software, la descripción del modelo puede ser incompleta, lo que dificulta la replicabilidad de los resultados, lo que lo vuelve difícil replicar las simulaciones en un entorno real.

Como se ha comentado en los capítulos anteriores, las simulaciones computacionales de humanos y, en este caso, de una expansión vírica, requieren una arquitectura capaz de juntar percepciones, motivaciones, toma de decisiones y la comunicación. Esta arquitectura será la individual del agente inteligente, y, en concreto, utilizaremos el modelo HUMAT [1], una arquitectura base desarrollada para la construcción de poblaciones artificiales, donde las cogniciones de los agentes, la toma de decisiones y las interacciones sociales se basan en la teoría científica social.

Un HUMAT no es más que una arquitectura, que define determinadas características para los agentes que la componen. La formación de las redes de individuos que comparten intereses es uno de los grandes puntos del proyecto, ya que la comunicación entre ellos influirá en la toma de decisiones (al final, la adherencia a la medida de contención definida). Por tanto, es muy importante definir como se organizarán los individuos entre ellos, así como sus características.

La arquitectura HUMAT formaliza una serie de procesos subyacentes a la dinámica social. El proceso de actuación de cada agente se puede dividir en varias etapas:

- Inicio.
- Evaluar las alternativas de comportamiento. En nuestro caso, aceptar la medida o no aceptarla.
- Seleccionar la alternativa preferida.
- Actuar.
- Efectos experienciales, esto es, los efectos que tiene la decisión del agente sobre el modelo, es decir, como se modifica la simulación a raíz de su selección de alternativas. Por otro lado, también tiene que ver con la experiencia del propio agente, que modificará su comportamiento si el efecto experiencial es muy grande.

- Actualizar la memoria y características del agente.
- Repetir desde el paso 2.

4.1 Evaluación de las alternativas de comportamiento

El inicio de la actividad del agente se basa en la satisfacción (o no) de necesidades y la medida en la que se satisfacen con las diferentes alternativas de comportamiento. Existen diferentes teorías sobre las necesidades que ofrecen taxonomías diferentes sobre lo que impulsa el comportamiento humano. En concreto, las categorías de necesidades que se utilizarán en el modelo son:

- **Necesidades experienciales:** Se refieren, entre otras, a comodidad y costes. Por ejemplo, la necesidad de poder mantener el trabajo durante un confinamiento completo, esto modifica la experiencial del agente en el sistema.
- **Necesidades sociales:** Se refieren a la necesidad de pertenencia a grupos sociales, afinidad a los mismos, etc. Es decir, sentirse cercano y aceptado/a por las personas importantes para uno. También incluye la seguridad social, el estatus social, etc.
- **Valores:** Se refiere a la autonomía y objetivos sociales.

Esta definición introduce una posibilidad de compensaciones entre diferentes grupos de necesidades, que pueden resultar en disonancias cognitivas que impactan, directamente, en el procesamiento de información y en la toma de decisiones final.

La **evaluación** de una alternativa de comportamiento vendrá definida, pues, por la satisfacción esperada para esa alternativa para cada una de las necesidades, ponderada por la importancia que el individuo le da a cada una de ellas. Esto puede dar lugar a que la satisfacción de una alternativa sea satisfactoria, neutral o no satisfactoria. Como se comentaba, cuando una alternativa evoca niveles suficientes de satisfacción e insatisfacción en uno (o más) grupos de necesidades, se da un estado de disonancia cognitiva, y el HUMAT se enfrenta a un dilema. Según el tipo de compensación entre los grupos de necesidades, se pueden identificar tres tipos diferentes de dilemas.

- **Dilema experiencial:** Ocurre cuando un comportamiento produce satisfacción en las necesidades y valores sociales e insatisfacción en las necesidades de experiencia, o cuando las necesidades de la experiencia se satisfacen y los valores y las necesidades sociales no se satisfacen. Por ejemplo, cuando el agente está considerando aislarse en su casa en un caso de confinamiento completo. El proceso interrumpe la vida del HUMAT (lo

que provoca insatisfacción en las necesidades experienciales), pero todos sus vecinos y amigos están de acuerdo en realizar actividades para el cumplimiento de las medidas, de forma que el agente se ve satisfecho en el ámbito social y también en sus valores en los que prima evitar la expansión.

- Dilema social: Ocurre cuando un comportamiento produce satisfacción en valores y necesidades experienciales pero insatisfacción en necesidades sociales, o cuando produce satisfacción en necesidades sociales e insatisfacción en valores y necesidades experienciales.
- Dilema de valores: Ocurre cuando un comportamiento produce satisfacción en las necesidades de la experiencia y las necesidades sociales e insatisfacción en los valores, o cuando produce satisfacción en los valores e insatisfacción en las necesidades sociales y necesidades experienciales.

4.2 Selección de la alternativa preferida

Una vez que el HUMAT ha evaluado el nivel de satisfacción de todas las alternativas de comportamiento disponibles, elige la más satisfactoria. Sin embargo, si las dos alternativas son igualmente satisfactorias, explorará "más a fondo" las opciones con respecto a cuánta disonancia cognitiva evocará cada una y elige la alternativa que requiere el menor esfuerzo para reducir la disonancia. Si aplicando este proceso, las alternativas siguen siendo similares, el HUMAT muestra una tendencia por el comportamiento que es más satisfactorio en las necesidades experienciales, de ahí que tienda a inclinarse hacia el hedonismo. Si aún así, las alternativas siguen siendo similares, elegirá de manera aleatoria entre las opciones.

4.3 Actuación y efectos experienciales

Si se ha tomado acción (una de las alternativas de comportamiento), pero la disonancia no se resolvió a un nivel suficiente, el HUMAT continuará investigando, en su red de contactos, información útil para la reducción de disonancias o sobre mejores alternativas. Esto es, preguntará a los individuos de su red social acerca de su toma de decisiones y necesidades, valores, etc. Por otro lado, si ha tomado la acción, también puede comunicar su opinión de comportamiento y selección o su opinión sobre las disonancias y alternativas a los individuos de su red, para intentar convencerlos de que lleguen a pensar de la misma forma que él, para reducir disonancias y centrarse en la alternativa seleccionada.

4.4 Actualización de la memoria y valores

Los HUMAT están "equipados" con representaciones cognitivas de las evaluaciones pasadas de posibles alternativas conductuales. La memoria de un HUMAT está representada por una pila de alternativas de comportamiento clasificadas de acuerdo con su capacidad de satisfacer múltiples necesidades. Cuanto más resultados positivos y menos negativos, mayor es la capacidad de satisfacción de necesidades de un comportamiento.

Modelos epidemiológicos

Dentro del modelo planteado, no solo se simulará el comportamiento de los individuos que forman la población gracias al HUMAT. También necesitamos modelar el comportamiento de la expansión del virus epidemiológico. Para ello, utilizaremos un modelo epidemiológico común y básico, llamado SIR, pero modificado para lograr adaptarlo mejor al caso de resolución planteado. El modelo se denomina SIR por la capacidad de los individuos que lo componen de pertenencia a tres diferentes grupos: Susceptible, Infected y Removed.

5.1 Modelos epidemiológicos

Este tipo de modelos se usan frecuentemente como instrumento para evaluar diversas actividades de gestión de las enfermedades. El valor de los modelos epidemiológicos reside en su capacidad de estudiar hipótesis y de hacer que los responsables de tomar decisiones sepan de antemano qué consecuencias tendrá la expansión vírica y qué impacto tendrán tanto las estrategias de control (medidas de contención) como la toma de decisiones del agente (la adherencia hacia estas medidas).

Teóricamente, la complejidad y variabilidad intrínseca de los sistemas biológicos, deberían limitar el uso de los modelos como instrumentos de predicción durante el brote real. Es decir, serán mucho más útiles antes o después del brote, en particular, para el análisis retrospectivo de brotes previos. Debido a que no tenemos casos previos de brotes similares, ejecutamos la simulación durante el mismo, utilizándolo como si ya hubiese sucedido, y simulando que volverá a darse.

Los modelos en sí, son solo un instrumento para obtener un asesoramiento científico y sus resultados deberían de ser evaluados junto con los datos obtenidos con estudios experimentales, la experiencia práctica y otros conocimientos científicos.

5.2 Modelo SIR

Como se ha comentado, para el caso planteado, se utilizará el modelo SIR [6], uno de los modelos epidemiológicos más simples capaces de capturar muchas de las características típicas de los brotes epidémicos. El nombre del modelo proviene de la existencia de tres grandes grupos a los que pueden pertenecer los individuos de la población simulada. El modelo relaciona las variaciones de las tres poblaciones a través de la tasa de infección el periodo infeccioso promedio.

Se definen diferentes estados que puede tener un individuo, definiendo así las etapas del desarrollo de la enfermedad:

- S: **Susceptible**: estado en el que el individuo todavía no ha sido afectado por el virus pero puede serlo. Son los individuos "sanos" que no han contraído la enfermedad.
- I: **Infectado**: estado en el que el individuo ha sido infectado por el virus.
- R: **Recuperado**: estado en el que el individuo fue afectado por el virus pero lo ha superado (y es inmune) o ha muerto a causa del mismo.

A pesar de pertenecer a un estado, un individuo puede moverse entre ellos del siguiente modo:

- Un individuo susceptible puede pasar al estado infectado, dada una probabilidad de contagio y dependiendo de los factores definidos en la implementación concreta del modelo. En concreto, y generalmente, suele darse cuando un individuo susceptible toma contacto con otro infectado.
- Un individuo infectado puede pasar al estado recuperado, que indicará inmunidad ante la enfermedad. Esto puede darse cada cierto tiempo predefinido y bajo cierta probabilidad de volverse inmune frente al virus.

Es importante definir ahora dos conceptos **de la simulación de expansión del virus** a tener en cuenta durante todo el proyecto:

- **Comportamiento temprano**: etapas iniciales de la expansión. Es importante tener el control y conocer como actuará el virus en esta etapa por dos motivos:
 - Se necesita tiempo para desarrollar vacunas e intervenciones médicas.
 - La mejor forma inicial para contener la pandemia será mediante la cuarentena y/o vacunación temprana.

- **Comportamiento tardío:** patrón de comportamiento de la pandemia en fases avanzadas. También es importante conocer este comportamiento y controlarlo, ya que permite medir (y por tanto, predecir) el alcance de la pandemia, los individuos infectados, etc.

Como puede observarse, estas definiciones asumen que todos los individuos tienen la misma probabilidad de entrar en contacto con un individuo infectado, de modo que se elimina la necesidad de conocer la red de contactos sobre la que se difunde la enfermedad, reemplazándola con la asunción de que cualquiera puede infectar al resto. Esta característica del modelo SIR se denomina **Mezclado Homogéneo**: En cada unidad de tiempo, cada individuo tendrá $\langle k \rangle$ contactos con otros individuos de la población escogidos aleatoriamente.

También obtenemos acceso a diferentes definiciones de medida y conceptos a tener en cuenta para la implementación de este proyecto[7]:

- Si en la población de tamaño N , hay I individuos infectados, y S individuos susceptibles en el instante t , la probabilidad de encontrar un individuo infectado será $i=I(t)/N$ y susceptible es $s=S(t)/N$, siendo $i+s=N$
- La enfermedad solo se produce cuando hay contacto entre un individuo infectado y otro susceptible. En cada instante de tiempo, un individuo infectado tiene contacto con un promedio determinado de individuos sanos, a los que transmite la enfermedad con cierta probabilidad (proceso denominado **contagio simple**).
- Como hay I individuos infectados, el ratio medio de nuevas infecciones en cada instante de tiempo t es: (probabilidad-contagio) * (promedio-contactos-diarios) * $(s) * (i)$.

El modelo SIR básico explicado ha sido modificado ligeramente para adaptarlo a la temática de este TFG. Estas modificaciones se detallarán en el capítulo dedicado a desarrollo.

Herramientas utilizadas

EN esta sección se muestra un compendio de las herramientas utilizadas durante el desarrollo del proyecto.

6.1 NetLogo

NetLogo [8] es uno de los entornos de modelado basado en agentes más utilizados. Además de proporcionar el entorno (IDE), es también un lenguaje de programación propio diseñado para el desarrollo de modelos basados en agentes, de los que ya hemos hablado.

Fue desarrollado por Uri Wilensky [9] basado en Logo. El mundo de NetLogo está formado por agentes, que pueden seguir instrucciones. Existen, en este mundo, cuatro tipos de agentes:

- **Tortugas:** Agentes que se mueven por el mundo y ejecutan instrucciones.
- **Parcelas:** El mundo es bidimensional y dividido en parcelas. Cada parcela es una pieza cuadrada de "tierra" sobre la que las tortugas se pueden mover.
- **Enlaces:** Existen entre dos tortugas, denominadas extremos. Si cualquiera de los dos extremos muere, el enlace también muere. Visualmente, se representa como una línea con información interna entre dos tortugas.
- **Observador:** No es un observador pasivo, es el que se encarga de dar instrucciones a los demás agentes. Además, no tiene ubicación en el mundo, puede imaginársele "mirando" al resto de agentes.

NetLogo incluye una amplia librería de modelos de ejemplo pre-programados, mediante los cuales puede estudiarse el funcionamiento tanto de la aplicación (interfaz y utilidades) como del lenguaje propio. Estos modelos de ejemplo están categorizados, por lo que algunos de ellos han servido para el desarrollo de este trabajo, como pueden ser: "Virus on a Network" o "epiDEM Basic".

En la figura 6.1 se sigue el proceso mediante el que se accede a la biblioteca de modelos de NetLogo.

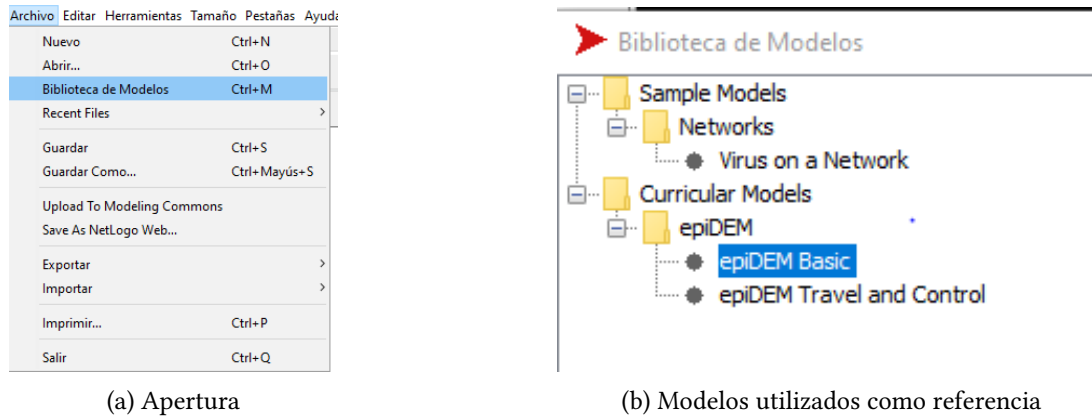


Figura 6.1: Biblioteca de modelos de Netlogo

Se ha elegido NetLogo como herramienta, no solo por sus propiedades ya mencionadas y porque es gratuita, sino por la sencillez para reproducir diferentes simulaciones sin tener que cambiar todo el código. Y es que la interfaz de NetLogo permite modificar variables del código de forma temporal mediante el uso de interruptores, deslizadores y más. Además, dentro de la propia interfaz, se aportan diferentes estilos de gráficas y tablas programables, para medir la simulación visualmente y con datos tanto en tiempo de ejecución como a posteriori para su análisis.

En la figura 6.2 se muestra la interfaz gráfica creada para el modelo "epiDEM Basic". Esta interfaz la define el desarrollador para el usuario, por lo que será diferente para cada modelo de la biblioteca.

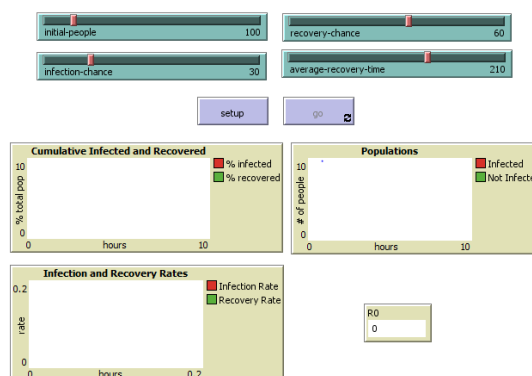


Figura 6.2: Ejemplo de interfaz para el modelo epiDEM Basic

6.2 Microsoft Teams

Se trata de una plataforma unificada de colaboración y comunicación proporcionada por Microsoft. Como el servicio se propone mediante el paquete de Office, se puede acceder a él desde la propia universidad. La herramienta fue publicada en 2017, y, como curiosidad, fue creada durante un Hackhaton interno de la propia empresa, al que, a posteriori, se le añadieron y modificaron elementos para crear el Teams que conocemos en la actualidad. La figura 6.3 hace referencia al logo de la herramienta.



Figura 6.3: Logo de Microsoft Teams

Esta herramienta nos ha dado la principal llave para la comunicación y organización entre alumno, tutores y colaboradores, proporcionando un medio mediante el cual podíamos mantener las reuniones programadas, tener una ágil conversación e ir depositando documentos de manera periódica, utilizándolo también, pues, como una especie de nube temporal.

6.3 Git

Git es un sistema de control de versiones gratuito y de código abierto, diseñado para manejar tanto pequeños proyectos como más grandes, con el mismo nivel de agilidad y sencillez. La figura 6.4 hace referencia al logo de la herramienta.



Figura 6.4: Logo de Git

Esta herramienta fue utilizada para mantener versiones limpias y funcionales del código de NetLogo, de manera que si en algún momento conviene volver atrás en estas versiones y recuperar alguna de ellas, sea sencillo. Además, proporciona seguridad, ya que sirve como sistema de guardado.

Desarrollo del proyecto

En este capítulo hablaré sobre el desarrollo del proyecto, su método, fases y características concretas.

7.1 Método de trabajo

La metodología de trabajo seguida durante todo el desarrollo del proyecto fue Scrum, ya que pueden proponerse iteraciones de trabajo con periodos cortos, lo que resulta muy apropiado dada la naturaleza de este TFG. En cada iteración, se seleccionarán objetivos para resolver con mayor prioridad, solventando así, poco a poco, todos los requisitos generales del proyecto, pero identificándolos paso a paso.

Estas iteraciones se realizaron mediante reuniones semanales en Microsoft Teams, donde se marcaban los objetivos a seguir y requisitos a cumplir para la siguiente semana. Además de esto, se revisaba el avance del proyecto con tutores y colaboradores, y, también, se solucio- naban dudas o bloqueos en las iteraciones, de forma que era posible que una iteración durase más de la semana planteada, consiguiendo que el desarrollo del proyecto tuviese una lineali- dad cómoda y sencilla.

Todo el código y los documentos relacionados con el proyecto se puede ver desde el Git- Hub público del proyecto [10], y se hará referencia a alguno de estos documentos durante este capítulo.

7.2 Sprint 1: Modelo Epidemiológico Simple

La primera fase del proyecto fue definir y modelar una primera aproximación simple de la transmisión del virus epidemiológico. Dadas determinadas propiedades, se define su ex-

pansión y el objetivo es simular la expansión de un virus en una sociedad de agentes sin inteligencia y sin conexiones entre los mismos. Es decir, únicamente se simula la expansión del virus, aunque con diferentes variables que el usuario puede modificar para ver la influencia que tienen en cuanto a la propia expansión. Las características principales de esta fase serán, por tanto, que los agentes no posean ningún tipo de inteligencia ni capacidad para tomar decisiones y que no existan, tampoco, relaciones entre los mismos.

7.2.1 Escenario inicial

El primer paso, entonces, fue crear un entorno en el que el virus pudiese expandirse. Para ello, se creó una sociedad de agentes, cuya única capacidad es ser infectado o no.

Para conseguir un escenario adecuado para esta fase, se proporciona un deslizador para seleccionar el tamaño máximo de la población. En este caso, el máximo de individuos dentro de la sociedad será de 100. En esta fase, nuestros agentes, a los que llamaremos HUMATS, únicamente poseerán una propiedad: el estado del agente. Este estado puede variar entre infectado y susceptible.

En la figura 7.1 puede verse la diferencia entre dos estados iniciales. Los individuos están representados con una forma "humana". Aquellos con color verde son individuos susceptibles a ser infectados, es decir sanos, mientras que los que tienen color rojo son individuos ya infectados. El usuario podrá modificar la cantidad de población en el escenario mediante el deslizador "initial-population", y mediante el botón "setup", colocará, de manera aleatoria, a los agentes a lo largo del escenario.

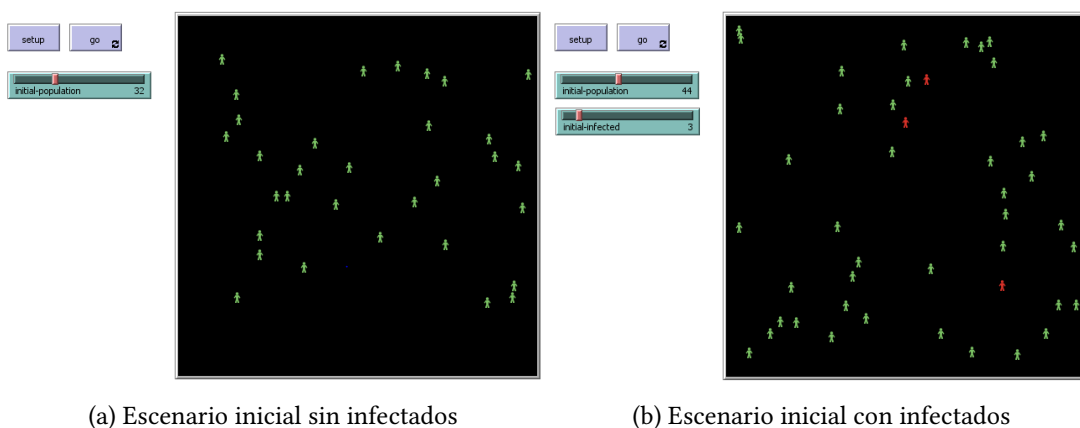


Figura 7.1: Escenario inicial

Se dispone de un deslizador en la interfaz, llamado "initial-inmfected" mediante el cual

seleccionaremos el número de infectados para la situación inicial. El máximo que puede tomar esta variable es de $(\text{initial-population} / 2)$. Una vez definida esta variable y el valor de la población, se colocarán en la pantalla, en rojo, aquellos agentes infectados (tantos como se haya seleccionado con la variable), y, aquellos que no estén infectados, se dibujarán en verde.

7.2.2 Implementación del modelo SI

El siguiente paso consiste en la implementación de la reducción del modelo SIR, lo que denominaremos SI, en la que no se tienen en cuenta los individuos recuperados. Esta aproximación nos permitió conseguir unos algoritmos de expansión simples para que la simulación comience sobre una base bien definida y capaz de ejecutarse sin complicarse demasiado.

Se controla el avance de los "días" mediante la variable <ticks-day> y se realizan los algoritmos de expansión del virus. Es importante remarcar que la simulación finalizará una vez que todos los individuos estén infectados, ya que no se tiene en cuenta la recuperación de los mismos.

Tendremos una función para determinar los contactos diarios de cada agente, que relega, a su vez, en dos funciones de expansión del virus: una simulará la expansión para aquellos humats con el estado infectado, y otra para los que tengan estado susceptible.

- En el caso de los individuos infectados, realizará n contactos con otros individuos de la simulación. Si el individuo con el que tiene contacto también está infectado, no ocurrirá nada. Sin embargo, si el otro individuo es susceptible, le contagiará la enfermedad con una probabilidad k , que, como comentaba, será definida por el usuario en la interfaz.
- En el caso de los individuos susceptibles, también realizarán el mismo número de contactos. Si lo hacen con individuos susceptibles, no ocurre nada, ya que están fuera de peligro. Sin embargo, si el contacto es con un individuo infectado, el primero quedará infectado bajo la misma probabilidad comentada con anterioridad.

Es necesario realizar diferentes puntualidades sobre estos algoritmos:

- Cuando un individuo infectado hace contacto con otro individuo infectado, a la vista de la simulación no sucede nada, ya que dicho contacto no es influyente.
- Cuando un individuo susceptible hace contacto con otro de la misma índole, tampoco sucede nada, ya que no es influyente (al no tener ninguno de los dos la enfermedad, no hay contagio en dicho contacto).
- El contagio se realiza mediante una función aleatoria. Se genera un número flotante entre 0 y 1 y, si es menor que la variable de probabilidad de contagio, se producirá el

contagio si están en contacto dos individuos con estado diferente (uno infectado y otro susceptible).

Para controlar el avance diario, se utilizará el "tick" interno de NetLogo, que representa 1 iteración de la función go. Para controlar los "días" de expansión del virus, se dispone de un nuevo deslizador que el usuario puede modificar, y, por tanto, la velocidad de la simulación. Este deslizador se llamará <ticks-day>, y, mediante incrementos de 50.000 ticks por día, puede llegar hasta 500.000 si se desea ver días con una expansión muy acelerada.

Tras las explicaciones anteriores, actualmente la interfaz de usuario tiene el aspecto de la figura 7.2. Como se puede ver en la misma, se dispone de diferentes botones y deslizadores. Los botones "setup" y "go" controlarán la configuración y ejecución de la simulación respectivamente, mientras que, por otro lado, los deslizadores ayudan a la configuración del mismo denotando: población inicial, número de infectados en primera instancia, número promedio de contactos diarios y la probabilidad de contagio.

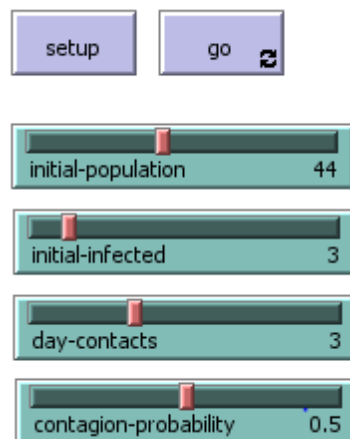


Figura 7.2: Estado de la interfaz

En este punto, se puede realizar una simulación con la implementación del modelo simple. Para ello, el usuario puede realizar una configuración de las variables de manera deseada, y tras pulsar <setup> verá los humats en pantalla. Para que la simulación tenga lugar, es necesario pulsar el botón <go>. Un resumen de la simulación que acabo de comentar puede verse en la figura 7.3.

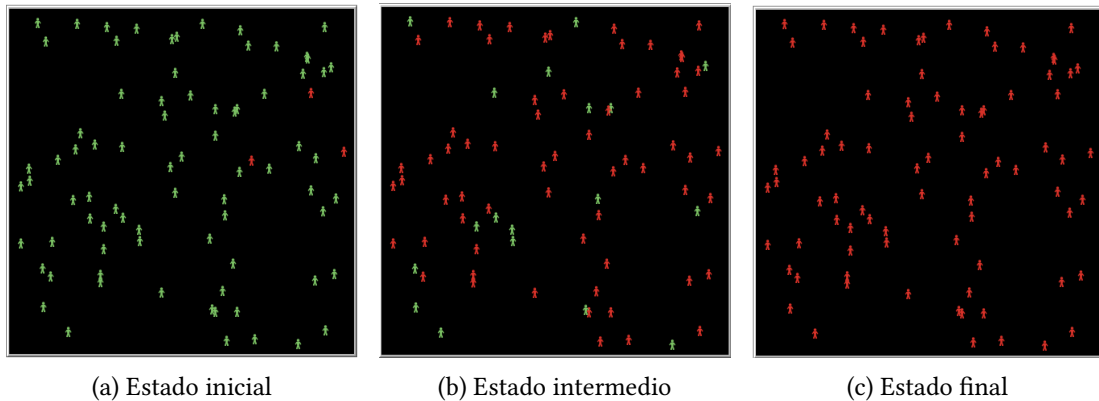


Figura 7.3: Simulación en la Fase 1 del desarrollo

7.2.3 Estudio de los resultados

El último paso de esta fase fue el de observar resultados de las simulaciones y valorar el funcionamiento de las mismas. Para ello, se agregaron gráficas y medidores para controlar el valor de las diferentes variables y valores durante y al finalizar la simulación:

- Contador de individuos cuyo estado es susceptible.
- Contador de individuos cuyo estado es infectado.
- Gráfico con los contador anteriores, para observar la curva de evolución de la pandemia.
- Tasa media de nuevos contagios en cada instante (cada día - cada determinados ticks).

Con estos visores, podremos analizar una iteración de la simulación, como la de la Figura 7.4

Como puede observarse en los monitores y gráficas resultantes de la simulación, el estado final resulta en todos los individuos infectados. Además, las curvas de las gráficas son predecibles y van a reproducirse de forma similar en todas las configuraciones para las variables de usuario. La razón de esto es el funcionamiento de la expansión: cada vez que un individuo hace una visita, tiene posibilidades de infectarse, y, aún infectado, continúa con las visitas, lo que produce una subida exponencial del número de infectados y una bajada igual de los susceptibles. En resumen, al principio sólo hay unos pocos individuos infectados, pero, al final, toda la población se afecta, aunque, como se puede ver en la gráfica de la tasa media de nuevas infecciones, cada día se infectan menos personas.

Esto resulta, además, en una gráfica de tasa media de nuevas infecciones también predecible, ya que cada vez será mayor hasta que, una vez que prácticamente toda la población sea infectada, baje de golpe por falta de individuos a los que infectar.

7.3 Sprint 2: Modelo SIR Completo

Tras haber definido una base para el proyecto (el modelo SI de la fase anterior) y haber tomado contacto con todas las herramientas que se utilizarían en el proyecto, el siguiente paso fue el de completar el modelo SIR por definición, es decir, añadir la posibilidad de recuperación/muerte en el modelo implementado, todavía sin conexiones entre los agentes del mismo.

Podemos distinguir, en esta fase, tres dinámicas a seguir: definición del modelo completo, implementación y análisis de resultados.

7.3.1 Definición del modelo SIR completo

El modelo SIR extiende al SI considerando la posibilidad de recuperación o eliminación de los individuos pero sin reinfección. Una vez que un individuo se infecta, puede quedar permanentemente inmune a la enfermedad o morir. En el desarrollo del proyecto, se modificará esta definición.

En esta aproximación, se aplica el mezclado homogéneo explicado en la fase anterior, pero esta vez con tres estados: S, I y R.

Los individuos infectados pasan periódicamente al estado R, lo que deja al modelo con, únicamente, dos transiciones entre estados posibles (S \rightarrow I) o (I \rightarrow R), como puede verse en la figura 7.5. En ellas, las variables colocadas sobre los cambios de estado determinan la probabilidad de cambio entre cada estado.

A efectos del modelo de expansión, el estado R representa tanto la inmunidad como el fallecimiento, ya que en ambos casos, el individuo deja de ser susceptible, es decir, no puede verse infectado.

La dinámica del proceso tiene dos fases. En la primera, los individuos infectados infectan a los susceptibles con (ratio de transmisión (probabilidad-contagio) * (promedio-contactos-diario)) siendo estas variables definidas por el usuario en la interfaz gráfica. En la segunda, los infectados se recuperan (o mueren) con una probabilidad de recuperación k , que también será definida por el usuario.

7.3.2 Implementación del modelo

El siguiente paso, una vez definido, es la implementación del modelo. Para eso, se realizaron diferentes cambios.

En cuanto a la interfaz:

- Se añade un deslizador (llamado <start-recovering>) para seleccionar el momento de

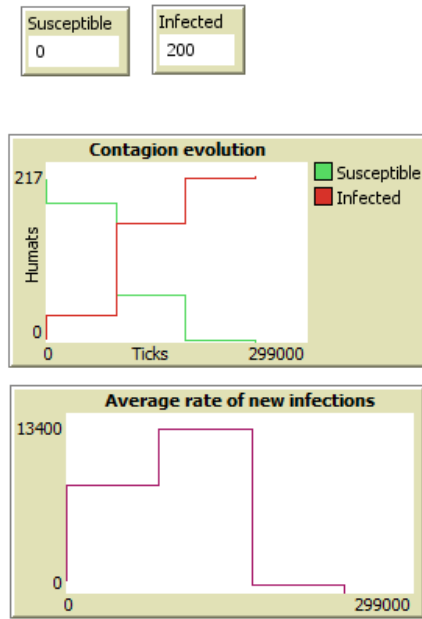


Figura 7.4: Resultados Fase 1



Figura 7.5: Transiciones Fase 2

la simulación en el que es posible comenzar la recuperación de los individuos. Lo cual permite que diferentes virus tengan diferentes tiempos de recuperación.

- Se añade un deslizador (llamado <recover-probability>) para poder seleccionar la probabilidad de cambio entre el estado I a R, entre los valores 0 y 1. Recordemos que, en la fase actual, este estado tiene tanto en cuenta recuperación como muerte, bajo la misma probabilidad.
- Se añade un monitor para controlar el número de individuos en estado R durante la simulación.
- Se modifican las gráficas de la fase anterior para tener en cuenta este nuevo estado y poder representarlo.

Estos cambios en la interfaz pueden verse en las figura 7.6.

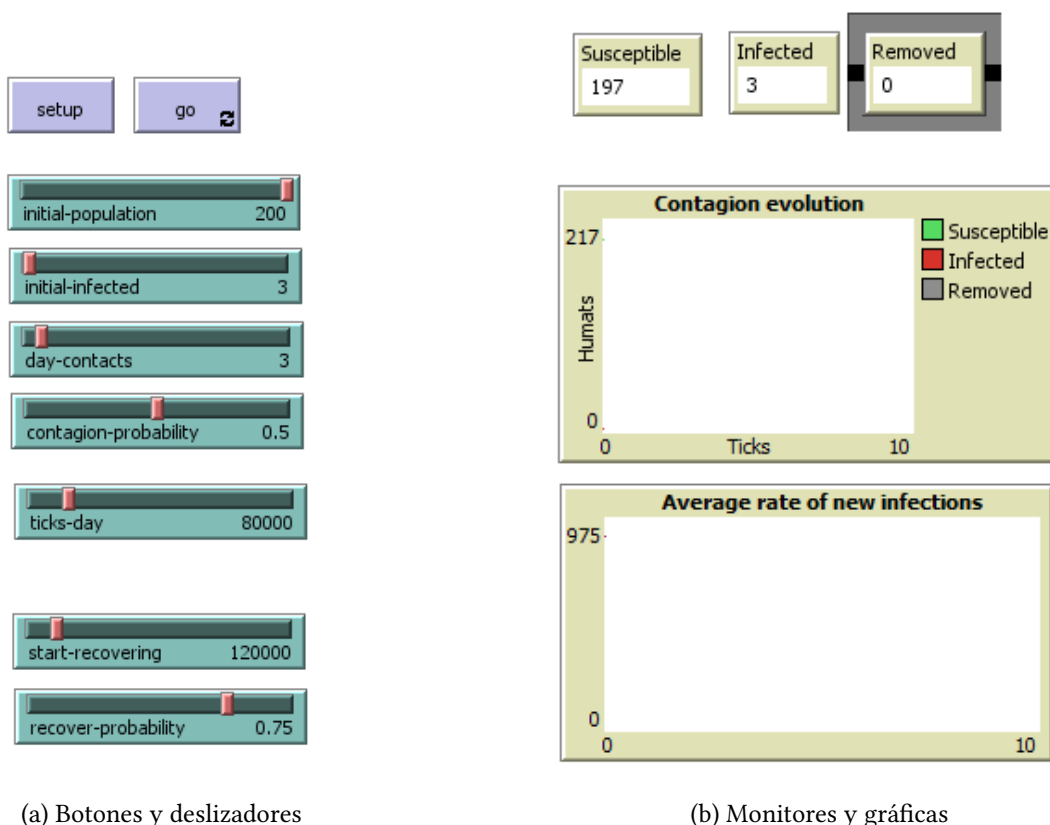


Figura 7.6: Interfaz Fase 2

Una vez definidas las nuevas variables e interfaz, se realizaron cambios en el código, teniendo en cuenta las nuevas transiciones entre estados y las posibilidades de infección, a

destacar:

- La nueva condición de parada será que no queden individuos infectados, lo que se busca es que todos acaben en estado S(susceptible) o R(removed).
- Además del punto anterior, en la función correspondiente al botón "go", se añadirá una comprobación. Si los ticks son mayores que los colocados por el usuario en la variable <start-recovering>, se llamará a la función "Update-Removed". Esta función se encarga de comprobar, dada la probabilidad de recuperación dada por el usuario, si el agente pasa al estado R. En principio, esto sucederá cada tick a partir de ese momento, aunque cambiará a lo largo del desarrollo del proyecto.
- En la función en la que se modela el contacto entre individuos no se tienen en cuenta aquellos individuos bajo el estado "Removed" ya que, al ser inmunes o estar muertos, no infectan ni se ven infectados por otros individuos.
- Se añade la función "Update-Removed", que se encargará de realizar la transición I -> R del modelo. Se le pide a todos los agentes con estado infectado que generen un número aleatorio entre 0 y 1. Si ese número es menor o igual a la variable <recover-probability>, el individuo pasará a ese nuevo estado. Si no, se mantendrá en el estado infectado.

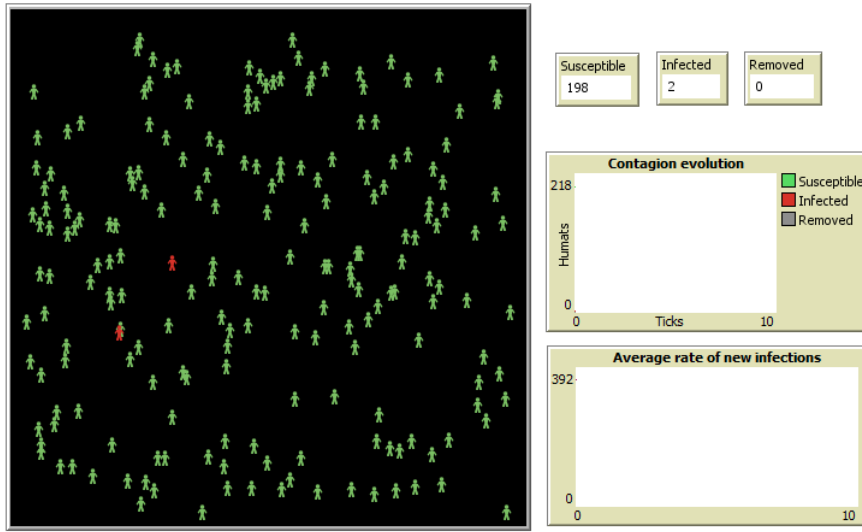
7.3.3 Análisis de resultados

Frente a los resultados de la fase anterior, en esta, a pesar de agregar únicamente un estado al modelo, ya se pueden obtener diferentes resultados dependiendo de como se planteen la ejecución de la simulación. Es decir, dependiendo de los valores que se le asignen a las diferentes variables de la interfaz, la simulación finalizará de un modo u otro, aunque siempre con un resultado similar.

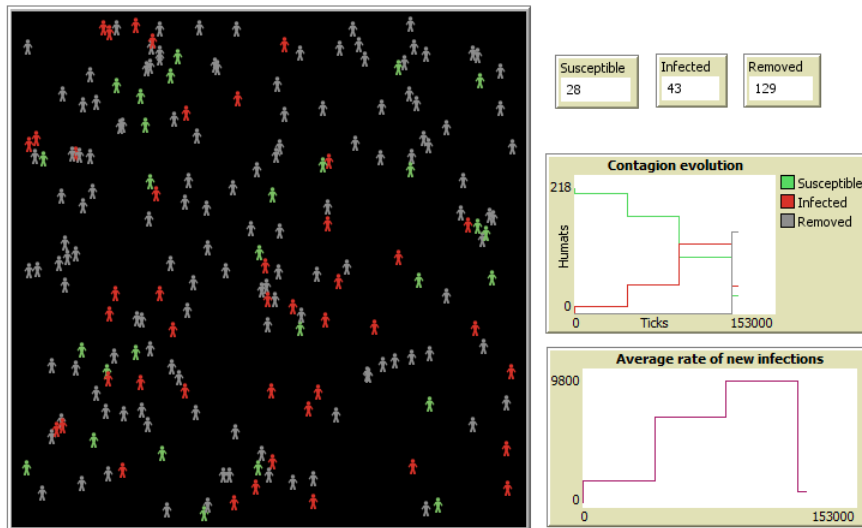
En la figura 7.7 se muestran los resultados de una simulación en la que se define una población de 200 individuos, de los cuales 2 poseen, inicialmente, un virus epidemiológico cuya probabilidad de contagio es del 30%, la probabilidad de volverse inmune o morir una vez contraída la enfermedad es del 70%, y los individuos únicamente podrán recuperarse una vez transcurridos dos días desde el inicio de la pandemia.

Como se puede observar en la figura, la expansión es rápida (debido a ese 30% de probabilidad de infección). Sin embargo, la recuperación también lo es, debido a su probabilidad y a comenzar en el segundo día.

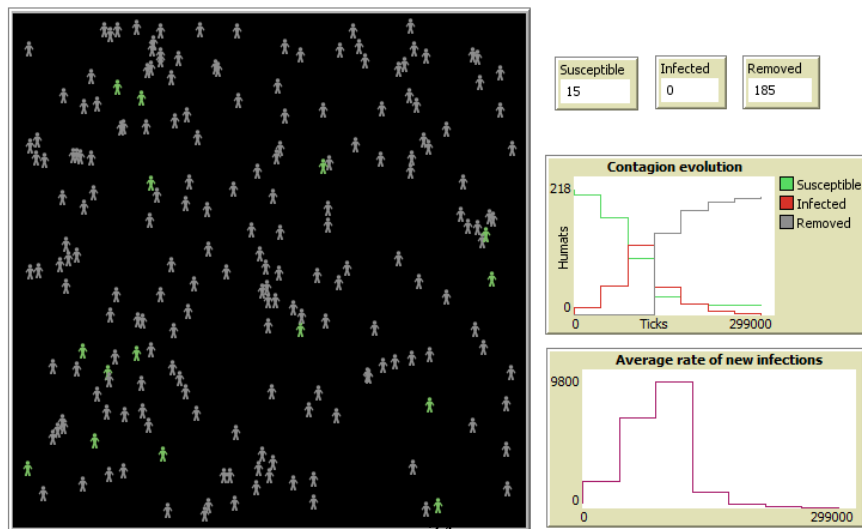
Si la tasa de contagio fuese del 10% y la recuperación a partir del quinto día, aunque con una mayor probabilidad del 80%, el resultado sería la de la figura 7.8.



(a) Estado inicial

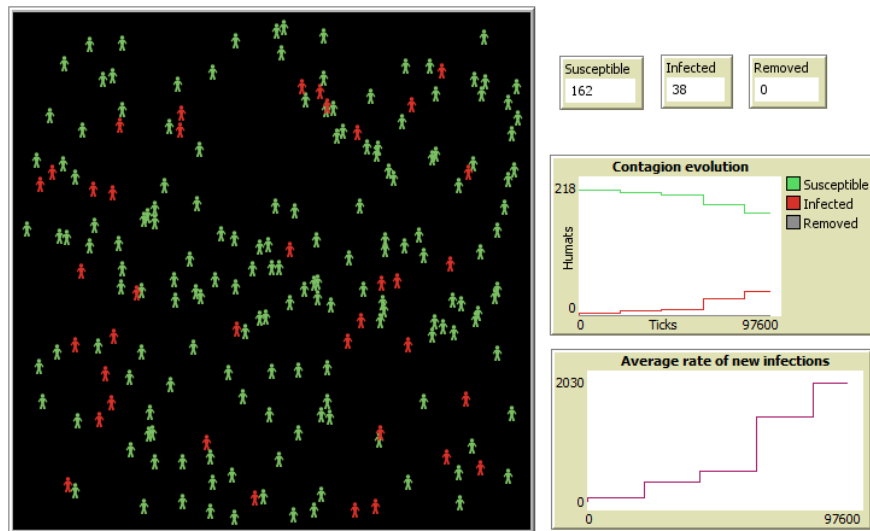


(b) Estado intermedio

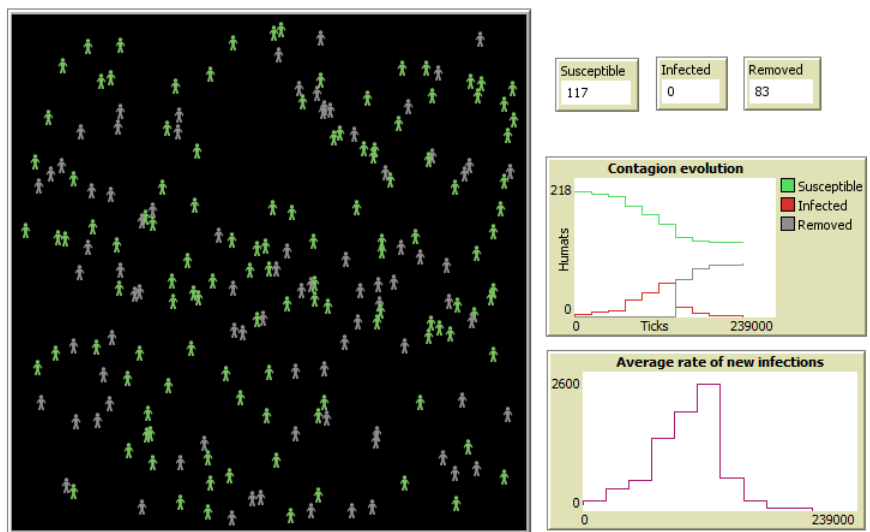


(c) Estado final

Figura 7.7: Simulación 1 en la Fase 2 del desarrollo



(a) Estado intermedio



(b) Estado final

Figura 7.8: Simulación 2 en la Fase 2 del desarrollo

En este caso, a diferencia del anterior, la expansión del virus fue mucho menor, debido a que la probabilidad de contagio más baja, por lo que fue más fácil pasar del estado I al R, o lo que es lo mismo, el número de infectados y nuevos infectados (como se ve en la correspondiente gráfica) se ve reducido de forma rápida a partir del quinto día.

Por tanto, en esta fase el proyecto permite simular virus de diferentes características y anticipar con cierta precisión el impacto que tendrá en una población pequeña. Sin embargo, todavía no se tienen en cuenta conexiones entre individuos, por lo que no es del todo fiable: se asume que cualquier individuo visitará a otro con la misma probabilidad, sea quien sea, de modo que es posible que el virus se extienda más rápido por toda la población en lugar de en zonas específicas de la misma.

7.4 Sprint 3: Incorporación de las conexiones

Para evolucionar el modelo SIR a un modelo en el que se pueda tener en cuenta el comportamiento social, es necesario implementar redes de conexiones entre los diferentes individuos. Estas redes sociales representarán familias, amigos/as u otro tipo de relaciones personales cercanas.

7.4.1 Implementación de la red

Para implementar la red, primero, es necesario que se generen las conexiones entre los agentes al inicializar el modelo, es decir, se añade una nueva función que se encargará de definir la red de conexiones entre agentes.

Además, se añade una nueva opción de configuración, llamada `<average-node-degree>` que se encargará de que el usuario pueda controlar el número de conexiones entre agentes.

En esta fase, la generación de la red de contactos sigue las siguientes condiciones.

- Para generar una red aleatoria y mantener una visualización comprensible, se genera un número de links igual a $(\text{average-node-degree} * \text{initial-population}) / 2$.
- La red es completamente aleatoria. Es decir, las conexiones serán entre individuos aleatorios en cada simulación.
- Las conexiones serán bidireccionales. Se entiende de aquí que la relación entre ambos es mutua y recíproca.

La visualización del modelo, una vez implementada esta red, es la que se muestra en la figura 7.9, donde se pueden observar las conexiones creadas entre los diferentes individuos de la red.

7.5 Sprint 4: Obtención y análisis de datos

Tras completar el modelo SIR en una red que genera relaciones sociales entre los agentes del propio modelo, para poder continuar con la implementación del proyecto planteado, es necesario obtener datos reales referentes al comportamiento social de la población frente a las medidas tomadas para frenar la expansión del virus.

7.5.1 Obtención de datos mediante encuesta

Para llevar a cabo esta recolección de datos, se decidió que lo más óptimo sería obtenerlos mediante una encuesta donde se midiesen tanto las características situacionales (lugar de residencia, tipo de vivienda, etc.) como las características psicológicas de la población en mitad de la pandemia generada por el COVID-19.

La encuesta incluida en el github con los documentos del proyecto [10] (encuesta.pdf), se realizó mediante la plataforma de Google Forms, con preguntas separadas en diferentes secciones;

- Situación vital (Género, Edad, Tipo de Hogar, Estado Laboral...)
- Eficiencia de medidas de confinamiento.
- Eficiencia de medidas de prevención posteriores al confinamiento.
- Grado de dificultad para acatar diferentes medidas frente a la expansión del virus.
- Grado de vivencia de diferentes experiencias durante el confinamiento referente a la salud física y emocional.
- Sentimientos durante el confinamiento.
- Acuerdo o desacuerdo con diferentes aspectos de las medidas de contención.
- Relaciones sociales antes y después del confinamiento.

La encuesta fue diseñada con el apoyo de la Profesora Adina Dimitru, psicóloga y miembro del grupo de investigación SEIPA de la UDC. El cuestionario tuvo un alcance considerable, obteniendo, en total, 2297 respuestas. Sin embargo, no todas estas respuestas son útiles para el modelo. Se difundió a través de medios electrónicos y redes sociales.

7.5.2 Limpieza de los datos

El siguiente paso de obtención de datos fue la limpieza de los datos, eliminando todas las respuestas no completas (que hayan dejado el cuestionario antes de finalizarlo) y aquellas

con indicios de trolling, ya que generarían datos falsos o estadísticas erróneas. Además, es importante seguir una convención para las respuestas de tipo libre, es decir, que todas las respuestas a una misma pregunta sigan el mismo formato (360m², 360 metros cuadrados, etc)

Tras limpiar el formulario, se obtuvieron 2035 respuestas útiles. Esto, a su vez, corresponderá con 2035 individuos en el sistema simulado. El análisis (tablas de frecuencias) de la encuesta está accesible en el GitHub del proyecto [10] (frequencies.pdf).

7.6 Sprint 5: HUMAT: Definición base y selección de alternativas

Una vez implementados los puntos anteriores, vino un sprint con una gran carga, ya que hubo que implementar la base del HUMAT, sus variables y su comportamiento base. Este modelo se desarrolló como parte del proyecto SMARTEES [1], en el que participan los directores de este TFG. Gran parte de este comportamiento ya está definido en la documentación disponible [1], por lo que se hizo una implementación en Netlogo, que más tarde se ajustó al comportamiento expansivo del virus, adaptando el modelo HUMAT al caso de este TFG.

7.6.1 Definición e implementación de las variables base

Lo más importante del HUMAT, como tal, son las variables que lo componen y que definirán el comportamiento de los agentes. Como se ha mencionado en el capítulo 4, este comportamiento estará definido por tres aspectos de necesidad: necesidad experiencial, necesidad social y valores.

Centralizando la información, se definen, como aspectos que determinan el comportamiento de un agente, los siguientes:

- Valor de importancia que el agente concede a las diferentes necesidades: Importancia de la necesidad social, importancia de la necesidad experiencial e importancia de los valores.
- Valor de satisfacción para cada una de las tres necesidades, es decir, dadas las circunstancias de un agente en qué medida se satisfacen sus necesidades.
- Valor de evaluación de cada una de las tres necesidades. Esto se define como la multiplicación de la satisfacción de una necesidad por la importancia que se le da a la misma, de modo que sirve para definir como el agente evalúa la satisfacción de esa necesidad.
- Un valor de satisfacción global, que junta las satisfacciones de cada una de las necesidades y se utiliza para comprobar con la red de conexiones del agente y para la selección

de alternativas de comportamiento.

Es importante destacar que estos aspectos se calculan para cada una de las dos alternativas de comportamiento : la aceptación o la no aceptación de la medida de contención planteada. En lo que sigue, la aceptación de la medida será la alternativa de comportamiento A, y la no-aceptación de la misma será la alternativa de comportamiento B.

7.6.2 Evaluación y selección de alternativas de comportamiento

Pues bien, la cuestión ahora es, ¿para qué servirán las variables que componen el HUMAT?. La respuesta ya se ha mencionado: servirán para calcular la aceptación de la medida de cada individuo. Dadas dos alternativas de comportamiento, elegirá aquella que le proporcione mayor satisfacción.

Una vez inicializadas estas variables, se utilizarán, para seleccionar la alternativa que más satisface a cada individuo. Para ello, se definen diferentes funciones y algoritmos que nos ayudan a ello:

- A la hora de inicializar las variables, se hacen los cálculos pertinentes:
 - Cálculo de la evaluación a partir de la satisfacción y de la importancia, a saber: $(\text{evaluación}) = (\text{satisfacción}) * (\text{importancia})$, para cada necesidad y alternativa de comportamiento.
 - Cálculo de la satisfacción global para cada alternativa de comportamiento como la media de todos los valores de satisfacción.
 - Cálculo de la aceptación de la medida: aquella de las dos satisfacciones que sea mayor, definirá la alternativa de comportamiento escogida por el individuo.
- Después de crear los individuos, se actualiza su satisfacción social. Para ello, se utiliza su red de conexiones de la siguiente manera: El individuo preguntará a sus seres cercanos (aquellos con los que tiene una conexión directa), si tienen una opinión similar o no-similar a la suya. Dependiendo del número de individuos cuya opinión sea similar, se cambiará su alternativa de comportamiento o no. Pero, ¿qué significa similitud de opinión? Se tomará como individuos con una opinión similar a aquellos que han optado por la misma alternativa de comportamiento. Por ejemplo, si un individuo ha aceptado la medida, aquellos de su red que hayan aceptado la medida se considera que tendrán una opinión similar a la suya.
- Gracias a este nuevo valor de satisfacción social, pueden actualizarse los valores de evaluación y de satisfacción global de las alternativas dadas las nuevas evaluaciones.

- Después de actualizar la satisfacción social y global mediante el uso de la red de conexiones del individuo, se volverá a seleccionar una alternativa de comportamiento, es decir, si se acepta/cumple la medida de contención o no. Es importante destacar aquí que esta decisión de alternativa se ampliará y focalizará un poco más en futuras fases del proyecto.

7.7 Sprint 6: Revisión del modelo epidemiológico

En este punto del desarrollo, se detectó que el modelo epidemiológico SIR utilizado hasta ahora no era suficiente para representar toda la información necesaria en este caso, siendo el objetivo de este sprint resolverlos.

7.7.1 Ampliación del modelo SIR

Lo primero en este punto es ampliar el modelo. Dados los datos del SARS-COV-2, sabemos que el virus puede reinfectar a personas. Por esto, no nos sirven solo tres estados, con dos transiciones del estilo: Susceptible -> Infectado -> Recuperado. Debemos ampliar el modelo para que permita más estados y transiciones entre los mismos. Para ellos, se definen los siguientes estados:

- **Susceptible:** Estado en el que el individuo puede contagiarse.
- **Infectado:** Estado en el que el individuo ha sido contagiado.
- **Inmune:** Estado en el que el individuo no puede volver a infectarse.
- **Muerto:** Estado de defunción del individuo.

Las transiciones entre estos nuevos estados pueden verse en la figura 7.10, definiendo los que siguen:

- **Susceptible a Infectado:** Un individuo susceptible puede infectarse si tiene contacto con un individuo infectado.
- **Infectado a Inmune:** Un individuo infectado puede volverse inmune a la enfermedad. Todavía podrá mantener contactos con su red de conexiones, pero no volverá a contagiarse nunca.
- **Infectado a Susceptible:** Un individuo infectado puede recuperarse y volver a ser susceptible. Es decir, que si vuelve a mantener contacto con un infectado, tiene una probabilidad de contagiarse.

- **Infectado a Muerto:** Un individuo infectado puede morir. Este caso es especial, ya que se elimina al individuo de la red.

7.7.2 Uso de la red de conexiones

En un sprint anterior se ha definido la red de conexiones entre individuos, sin embargo, no se ha llegado a usar todavía salvo para el cálculo de la satisfacción social. Uno de los usos la red de conexiones será mejorar el algoritmo de expansión, proceso que se aborda en este sprint.

Para ello, cambiaremos la definición del modelo SIR. Este definía los contactos como un número de enlaces completamente aleatorios entre los individuos de la población simulada. Si tenemos una red social, esta definición carece de sentido, por lo que proponemos una nueva.

- Se considera que se ha producido un **contacto** cuando dos individuos se juntan a menos de metro y medio durante 10 minutos o más.
- Este contacto no es aleatorio. Es más probable entablar contacto con alguien de la red social: hay más posibilidades de permanecer hablando un buen rato y sin distancia de seguridad con un amigo o compañero de trabajo que con un extraño por la calle.
- Esta definición no es excluyente, es decir, los individuos pueden seguir teniendo contacto con individuos aleatorios de la red. Un evento real para este caso sería, por ejemplo, algún problema en el supermercado que obligue a estar cierto tiempo en el lugar y sin cumplir todas las restricciones básicas (distancia social, tiempo limitado, etc.)

Para simular los contactos, se generarán para cada individuo números aleatorios flotantes entre 0 y 1, y que se comparan con la probabilidad (en este caso alta, entorno al 0.8) de contacto con un individuo de su misma red social. Esto quiere decir que todos los números generados por debajo del 0.8 indicarán un contacto entre los individuos.

7.7.3 Implementación de las medidas de contención

El siguiente paso importante es implementar las medidas de contención en el modelo de expansión vírico. Para ello, tomaremos tres grandes medidas, muy dispares, que nos ayudarán a plantear los casos más generales.

Sin medida de contención: es la medida más frecuente al inicio de la expansión, así como normalmente, al final (cuando quedan pocos casos).

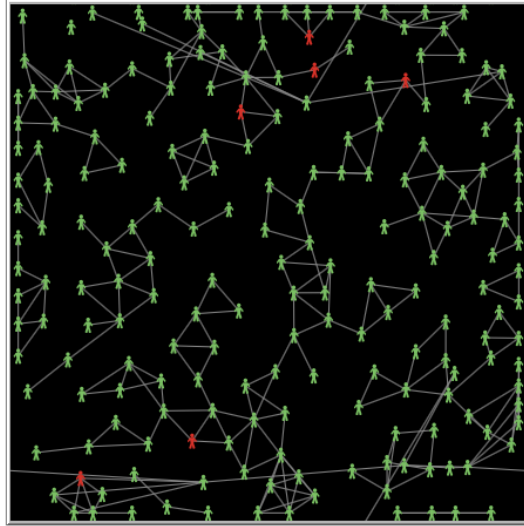


Figura 7.9: Red inicial

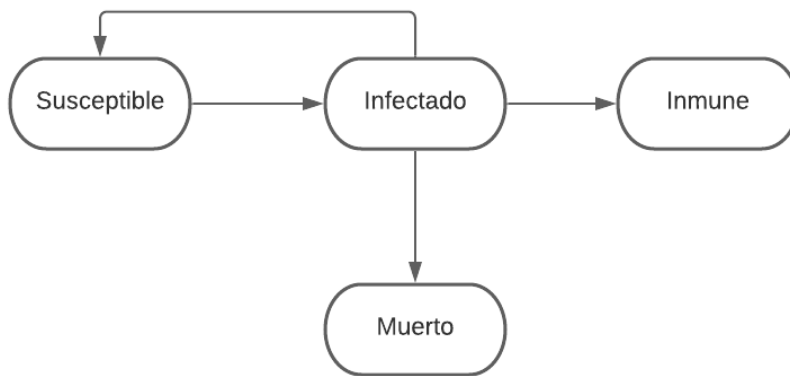


Figura 7.10: Diagrama de estados del modelo SIR ampliado

En este caso no se modifica en ningún aspecto el modelo de expansión, ya que no influye en los contactos ni en la propia expansión. Durante un periodo en el que no hay medidas, el comportamiento de la población será el "normal".

Confinamiento completo: el caso principal y más extremo de medida de contención. En este caso, se restringen todos los contactos, por lo que, en teoría, los individuos no podrán verse ni tener contacto con el que poder expandir el virus. Sin embargo, esto no es cierto para dos casos: aquella gente que trabaja incluso durante un confinamiento completo y que tiene que tener contactos sociales por el motivo que sea, y aquellas personas que simplemente no cumplen la medida de contención. Por ello, se añaden dos componentes a la expansión del virus.

- A la hora de realizar la expansión, si el individuo no acepta la medida (calculado mediante la arquitectura HUMAT), realizará los contactos igualmente. Sin embargo, solo podrán tener contacto dos personas que no cumplan la medida.
- Se añade una variable que permite que haya una pequeña posibilidad de contagiarse igualmente, aunque se cumpla la medida. Esta es ínfima, ya que solo aplica a aquellos individuos que trabajan en estos casos, y, como comento, es muy raro que no cumplan las restricciones básicas (distancia social, mascarilla, etc) si cumplen la medida general.

Confinamiento parcial: este caso engloba varias estrategias de confinamiento parcial. Aquellas en las que se permiten salidas de trabajo, de deporte, distanciamiento social, medidas preventivas en espacios cerrados, etc. La idea del proyecto, a futuro, es parametrizar estos casos, de forma que sea posible ampliar el abanico de medidas que se proponen. En este caso, aquellas personas que cumplan la medida tendrán muchas menos posibilidades de hacer contacto con un individuo (ya sea de su red social o no), mientras que aquellos que no la cumplen realizarán los contactos con normalidad, de forma que tienen una mayor probabilidad de contagiarse.

7.7.4 Ajustes menores

Después de realizar estos cambios, se ajustan diferentes componentes del proyecto para que tengan algo más de sentido, a tener en cuenta:

- Cada paso de la simulación se corresponde con un día.
- Se modifica el concepto de "tiempo de recuperación" que hasta el momento era aquel valor definido por el usuario que marcaba la cantidad de días que tenían que pasar hasta que todos los agentes pudiesen recuperarse de la infección. Para cada agente, se controlará, ahora, cada x días (los definidos por el usuario), si se ha curado.

- Se añade una probabilidad de defunción que el usuario podrá cambiar. De este modo, cuando cada agente llega a ese punto en el que se observa si se ha recuperado, también se observa si se ha muerto.
- En este punto del desarrollo estas son las variables que podrá proporcionar el usuario para parametrizar el sistema:
 - Número inicial de infectados.
 - Número de contactos diario que mantiene cada agente.
 - Probabilidad de contagio.
 - Frecuencia para la observación de aquellos individuos infectados.
 - Probabilidad de inmunidad.
 - Probabilidad de muerte.
 - Seleccionador de la medida de contención que se aplicará en el sistema.

7.8 Sprint 7: Mejora de la red de conexiones y salida del sistema

Dado que el sprint anterior fue dedicado a la mejora y revisión del modelo epidemiológico, es importante también mejorar la red de conexiones, de forma que sea mediante un algoritmo algo más complejo y extendido que, simplemente, mediante conexiones aleatorias. De la misma forma, es importante retomar la visualización de la salida del sistema, actualizando las gráficas y monitores para poder realizar los análisis correspondientes.

7.8.1 Red social

Existen diferentes modelos genéricos que se pueden usar para construir una red social mediante conexiones entre diferentes nodos (agentes en este caso). El algoritmo elegido en este caso fue Small Worlds, lo que genera una red cuyos nodos no tienen porque ser vecinos entre sí, pero que, en general, se puede llegar a cualquier nodo recorriendo todas las conexiones mediante un pequeño número de saltos. Para este tipo de redes, el coeficiente de agrupamiento es alto, lo que da como resultado un mundo de gente "desconocida" agrupada por pequeños grupos de gente "conocida".

En concreto, se utilizará una variante del algoritmo de Small Worlds, denominada Watts-Strogatz model [11], un modelo que es capaz de generar grafos no-direccionales (perfecto para nuestro caso) con las propiedades siguientes:

1: A cada individuo, se le relaciona con los cuatro individuos más cercanos. Esta relación es la más directa, y en un entorno pequeño, representa varios grupos, como la familia o amigos.

2: Dada una probabilidad de recableado, es decir, de cambiar una de esas relaciones, se puede cambiar una de estas relaciones por otra aleatoria.

Utilizando estos algoritmos, se logra una red de conexiones mucho más compleja que la tratada con anterioridad, como se puede observar en la figura 7.11.

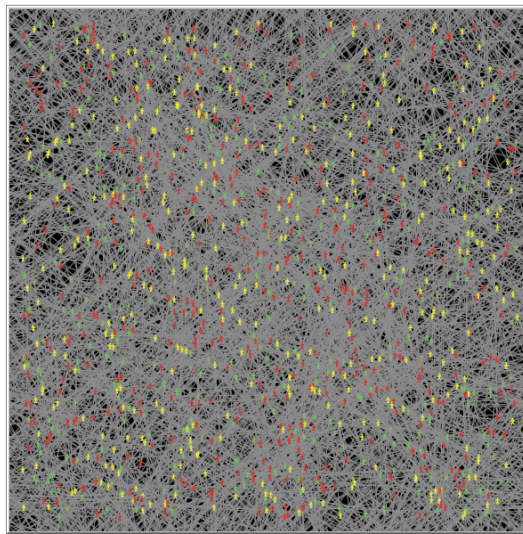


Figura 7.11: Visualización de las conexiones a mitad del proceso de expansión del virus

7.9 Sprint 8: Completando la arquitectura HUMAT

En este sprint pasa a ser el foco de atención la compleción del modelo HUMAT.

7.9.1 Añadiendo un nuevo grupo de necesidades

Debido a los datos obtenidos de la encuesta, se hace necesario implementar una necesidad más al grupo que define la arquitectura HUMAT. En este caso, será la necesidad de relación social. Debido al nombre, puede que parezca similar a la satisfacción social, sin embargo, no es así. La satisfacción social hace referencia a la condición de pertenencia al grupo del HUMAT, y, por tanto, marca el grado de satisfacción que determina la selección de alternativa de comportamiento de un agente en concreto frente a la de los demás agentes de la red social. La necesidad de relación social define directamente una nueva necesidad, la cual puede estimarse

mediante los valores de la encuesta y que define el peso que se le da a las conexiones de la red social.

7.9.2 Implementación de las disonancias

En el capítulo correspondiente al HUMAT se habló de las disonancias, que se producen cuando una alternativa de comportamiento genera niveles similares de satisfacción e insatisfacción en distintos aspectos.

Para llevar a cabo esto, tenemos que incluir las variables que definirán este comportamiento como características directas del agente. En concreto:

- Tolerancia a las disonancias: Este es un nivel de tolerancia a las disonancias que marcarán a partir de qué punto empieza a afectarle a un agente las disonancias a las que llega debido a los diferentes grupos de necesidades.
- Disonancia para cada alternativa de comportamiento. En este caso, serán "disonancia-A" y "disonancia-B", ya que le habíamos llamado A y B a las alternativas, siendo estas, respectivamente, aceptar las medidas de contención o no hacerlo.
- Peso de la disonancia. Si este peso es mayor a 0, indicará que hay una disonancia con un grupo de necesidades como se explicará con un poco más de detalle a continuación.
- Un valor booleano para diferenciar el tipo de dilema al que se enfrenta el agente. Habrá uno por cada grupo de necesidades (experiencial, social, valores o de relaciones sociales).

Ahora bien, ¿para qué sirven todas estas variables y cómo funcionan?. A la hora de realizar la configuración inicial del sistema se siguen los siguientes pasos:

- 1: Crear la población de 2035 agentes y asignar los primeros infectados.
- 2: Crear la red social (mediante el algoritmo de conexiones ya expuesto).
- 3: Actualizar la satisfacción social.
- 4: Actualizar las disonancias.
- 5: Decidir la alternativa de comportamiento.
- 6: Actualizar nuevamente la satisfacción social.
- 7: Actualizar nuevamente las disonancias.

El sexto y séptimo pasos son necesarios porque, una vez que seleccionamos la alternativa de comportamiento, la satisfacción social va a cambiar, y también las disonancias dependiendo de la alternativa que hayan seleccionado aquellos que forman la red social del agente.

La actualización de las disonancias se lleva a cabo mediante el siguiente algoritmo que se aplica para cada alternativa de comportamiento:

- 1: Se crea una **lista de evaluaciones**, que contendrá el valor de evaluación de cada grupo de necesidades para la alternativa de comportamiento concreta.
- 2: Se suma las evaluaciones positivas (**satisfactorias**) y negativas (**no satisfactorias**).
- 3: Se calcula el **valor de disonancia** para esa alternativa de comportamiento de la forma que sigue:
 - Se calcula el valor disonante como el mínimo de tanto los satisfactorios como los no satisfactorios.
 - Se calcula el valor consonante como el máximo entre los mismos valores.
 - Si la suma entre el valor disonante y el consonante es 0, el valor de disonancia será: $2 * \text{<valor disonante>}$. Sin embargo, si la suma es diferente de 0, el valor de disonancia será: $(2 * \text{<valor disonante>}) / (\text{<valor disonante>} + \text{<valor-consonante>})$.
- 4: Una alternativa de comportamiento evocará la necesidad de reducir una disonancia si esa disonancia supera el umbral de tolerancia de la misma. Este umbral de tolerancia se define, a la hora de crear al agente, como un número flotante aleatorio entre 0.5 y 1. Debido a esto, se calcula el **peso de la disonancia** como: $(\text{<valor de disonancia>} - \text{<umbral de tolerancia>}) / (1 - \text{<umbral de tolerancia>})$. Este valor de peso no lo utilizaremos en esta función, pero es importante definirlo como una característica del agente para posteriores operaciones.
- 5: Se comprueba aquellos grupos de necesidades en los que el agente se enfrenta a un dilema. Para ello, se sigue la tabla del cuadro 7.1. Como se puede ver en el mismo, el agente tendrá una disonancia en un grupo cuando para ese grupo la satisfacción es positiva pero para el resto lo es negativa, y al revés, cuando para ese grupo la satisfacción es negativa pero para el resto es positiva.

7.9.3 Mejora en la selección de alternativas

Hasta ahora se utilizaba la satisfacción como único factor a la hora de seleccionar una alternativa de comportamiento. Sin embargo, esto puede ajustarse ligeramente, añadiendo un

	Experiencial	Social	Valores	Relación Social
Experiencial	+	-	-	-
Social	-	+	+	+
Valores	-	+	-	-
Relación Social	+	-	+	+
	-	-	-	+
	+	+	+	-

Tabla 7.1: Tabla de disonancias

par de detalles para hacerlo más profundo y acercarlo a la realidad.

Primero, la decisión depende del nivel de satisfacción hacia cada alternativa, del mismo modo que antes. Si el valor de similitud es parecido en ambos casos, es decir, si este valor es menor a 0.2 (10% del rango de satisfacción teórico $[-1; 1]$), se pasará la decisión al siguiente nivel.

Segundo, si se llega a este nivel, la decisión dependerá del valor de disonancia que evoca cada alternativa. El agente escogerá la alternativa que le proporciona un menor nivel de disonancia. Si la diferencia entre los valores para las dos alternativas es menor al 0.1 (10 % del rango de disonancia teórico $[0; 1]$), se pasará la decisión al siguiente nivel.

Tercero, si se llega a este nivel, la decisión dependerá de la necesidad experiencial y su evaluación, dándole prioridad a aquella alternativa que proporciona una mayor evaluación de esta necesidad. Si la diferencia entre los valores para las dos alternativas es menor al 0.2, se pasará la decisión al siguiente nivel.

Cuarto, si se llega a este nivel, la decisión será completamente aleatoria. Se entiende que el individuo al que simula el agente no es capaz de tomar una decisión, por lo que la selección la hace de manera aleatoria.

En este punto, ya está completa la función de configuración de la población inicial. Antes de empezar la simulación, todos los agentes habrán decidido que alternativa de comportamiento utilizar, y, también, habrán actualizado sus valores de satisfacción social y sus disonancias.

7.9.4 Preguntas y señales

Para acercar el modelo todavía más a la realidad en la medida de lo posible, falta implementar las funciones de pregunta y señal en los agentes, definidas en la arquitectura HUMAT. Estas funciones se ejecutarán cada tick (día), antes de realizar las funciones de contactos y expansión, es decir, antes de aplicar el modelo epidemiológico.

La utilidad de estas dos funciones, tanto la de pregunta (a la que a partir de ahora me referiré como **Inquire**) como la de señalización (a la que me referiré como **Signal**), es la de influir en las decisiones de los individuos a los que está conectado al agente que las ejecuta.

En concreto, si el agente tiene un dilema del tipo experiencial, de valores o de relación social, ejecutará la función **Inquire**, buscando consejo en su red social para ajustar su "pensamiento", es decir, para reducir su nivel de disonancia.

Por otro lado, si el agente tiene un dilema social, ejecutará la función **Signal**, que le llevará a intentar convencer a sus allegados de que su selección es la correcta, intentando así disminuir su nivel de disonancia.

Para poder implementar todo esto, son necesarias un par de actualizaciones previas. El agente incluirá en sus características nuevas variables, que definirán la medida de confianza en su red social (valor de confianza) y dos valores para controlar cuándo está preguntando a otros agentes o cuándo está intentando convencer a otros agentes. Por otro lado, añadimos características a las conexiones de la red social, de modo que guarden información de este proceso. Se añaden valores como: si entre esos agentes ya se han preguntado/contactado, si han seleccionado la misma alternativa de comportamiento, un valor de persuasión y otro de credulidad. De este modo, obtendremos todas las variables necesarias para ejecutar el modelo.

Además, se necesita también un modo para actualizar las evaluaciones del individuo ya que se modificarán directamente al tener contacto con otros individuos. Lo que se consigue con este proceso de actualización de evaluaciones es que la satisfacción social se actualice conforme a las comunicaciones planteadas y de acuerdo al siguiente algoritmo:

1. Se calcula una lista ordenada con las relaciones que salen del agente. Como son no-direccionales, esto incluye también a las relaciones que llegan al agente. Esta lista estará ordenada por:
 - Primero, ascendentemente por aquellos agentes a los que todavía no se les ha preguntado.
 - Segundo, descendentemente por aquellos agentes con la misma alternativa de comportamiento seleccionada.

- Tercero, descendientemente por aquellos agentes cuyo valor de persuasión es mayor.
2. Se define el agente preguntado como aquel agente que es primero en la lista calculada.
 3. Se calculan las similitudes en la importancia dada para cada grupo de necesidades y para cada alternativa de comportamiento del modo que sigue: Si dos agentes valoran del mismo modo una necesidad (importancia de la misma) para una alternativa en concreto, el agente que influencia afectará al influenciado en un máximo del 40%. Se marca este máximo dado que siempre va a influir más una opinión personal frente a la de otra persona. Si dos agentes no tienen una similaridad en ese campo, un agente no influirá al otro.
 4. Se calculan las persuasiones para cada grupo de necesidades y alternativa del modo que sigue: (importancia-necesidad) * (valor-confianza). Como ya he comentado, este valor de confianza será del agente propio, y afectará a la persuasión para toda la red.
 5. Se hace una actualización de las satisfacciones como sigue: $(1 - (\text{persuasión del agente que influencia sobre el influenciado})) * (\text{satisfacción de las medidas del agente influenciado}) + (\text{persuasión del agente que influencia sobre el influenciado}) + (\text{satisfacción de las medidas del agente que influencia})$.
 6. Se realiza una actualización de las evaluaciones, disonancias y una selección de alternativa de comportamiento para el agente tratado.
 7. Se calcula la persuasión perteneciente a la relación (link) como la suma de todas las persuasiones previamente calculadas.

Como vemos, con la función de Inquire se consigue realizar una actualización de las evaluaciones, satisfacciones y selección de alternativa del agente por medio de preguntas a otros agentes, de modo que funciona según el objetivo esperado.

Por otro lado, tenemos la función de Signal, que se encargará de hacer que otros agentes tomen la misma decisión que el que la ejecuta, de forma que este último reduzca su nivel de disonancia. Como las funciones son similares, no se explicará con detalle que coinciden con el caso anterior.

1. Se calcula una lista ordenada con las relaciones que salen del agente. Como son no-direccionales, esto incluye también a las relaciones que llegan al agente. Esta lista estará ordenada por:

- Primero, ascendentemente por aquellos agentes a los que todavía no se les ha contactado.
 - Segundo, descendentemente por aquellos agentes que no tienen la misma alternativa de comportamiento seleccionada.
 - Tercero, descendentemente por aquellos agentes que son más fácil de persuadir.
2. Se calculan las similitudes en la importancia dada para cada grupo de necesidades y para cada alternativa de comportamiento.
 3. Se calculan las persuasiones para cada grupo de necesidades y alternativa.
 4. Se hace una actualización de las satisfacciones del mismo modo que en el proceso de Inquire.
 5. Se realiza una actualización de las evaluaciones, disonancias y una selección de alternativa de comportamiento para el agente tratado.
 6. Se actualiza la "credulidad" definida del agente y su contacto.
 7. Se realiza una actualización de las evaluaciones, disonancias y una selección de alternativa de comportamiento para el agente al que se le comunica.

En este punto el modelo ya es capaz de simular un comportamiento humano, tanto antes de comenzar la simulación (a la hora de generar las configuraciones, la red social y mediante la selección de alternativas y actualizaciones de satisfacciones en base a la red social) como durante la misma gracias a las funciones de Inquire y Signal, que simulan un comportamiento de comunicación muy dado en casos similares al planteado.

7.10 Sprint 9: Añadiendo los datos

Una vez implementado todo el modelo y probado su funcionamiento con datos aleatorios, es momento de añadir los datos obtenidos de la encuesta real. Para ello, son necesarios procesos importantes, como el tratamiento de los datos o la inclusión de los mismos en NetLogo, sin olvidarnos de la definición y organización de los mismos.

7.10.1 Organización de los datos

Para poder utilizar los datos, primero hay que definir dónde y cómo se van a utilizar. Para ello, se deben organizar los mismos en diferentes grupos dependiendo del campo donde se utilizarán. Estos grupos son los siguientes:

- **Demografía:** Datos que se utilizarán como variables representativas de los agentes. Entre estos datos podemos encontrar:
 - Edad
 - Género
 - Tipo de hogar, referente a: hogar unipersonal, pareja con hijos, pareja con hijos menores de edad, etc.
 - Tipo de vivienda, referente a: ciudad de más de 250m², campo, etc.
 - Estado laboral.
 - Si el trabajo que realiza el agente es esencial o no lo es.
 - Ingresos netos mensuales.
- **Satisfacción experiencial:** Aquellos datos que definirán las características de la satisfacción experiencial. Incluye preguntas, todas referentes al grado de dificultad de cada acción, como:
 - Superar un confinamiento completo.
 - Salir a trabajar durante el confinamiento.
 - Tener que trabajar desde casa.
 - No poder trabajar ni en casa ni fuera.
 - No poder realizar actividades al aire libre.
 - No poder realizar actividades en espacios cerrados.
 - Tener que pasar controles de temperatura para viajar o acudir a diferentes lugares.
 - Utilizar equipamientos de protección (mascarillas, guantes, etc).
- **Importancia experiencial:** Aquellos datos que definirán la importancia que se le da a la necesidad experiencial. Incluye preguntas como el grado de acuerdo con la siguiente afirmación: Respetar las medidas me protege contra el contagio.
- **Satisfacción relación social:** Aquellos datos que definirán las características de la satisfacción de relaciones sociales. Incluye preguntas, referentes también al grado de dificultad, como:
 - No poder desplazarse para visitar familiares.
 - No poder reunirse con amigos.
 - Tener que aislarse de otros familiares que viven en casa con el individuo, en el caso de contagiarse.

- Tener que mantener la distancia social en todas las interacciones fuera de casa.
- **Importancia social:** Aquellos datos que definirán la importancia que se le da a la necesidad social. Incluye preguntas sobre el grado de acuerdo con la siguiente afirmación: Hay mucha presión social para respetar las medidas de prevención del contagio.
- **Satisfacción valores:** Aquellos datos que definirán las características de la satisfacción de valores. Incluye preguntas como el acuerdo frente a la afirmación: En general, la gente respeta a rajatabla las medidas para frenar el contagio.
- **Importancia valores:** Aquellos datos que definirán la importancia que se le da a los valores. Incluye preguntas que miden el grado de acuerdo con diferentes afirmaciones, como:
 - Respetar las medidas de prevención de contagio es un deber cívico.
 - Respetar las medidas de prevención es un acto solidario.

Como puede observarse, no se incluye la satisfacción social, ya que esa no depende de los datos de la encuesta, sino de los cálculos del propio modelo.

7.10.2 Tratamiento e importación de los datos

Para poder utilizar los datos obtenidos en el modelo, con la herramienta de Netlogo, deben de pasar un pequeño tratamiento. Este tratamiento se encargará de preparar los datos para poder utilizarlos, y de juntarlos y transformarlos para agruparlos según lo mencionado en la subsección anterior.

El formulario realizado permite descargar las respuestas en un formato excel, en el que se ha realizado la limpieza de la que ya he hablado. A partir de esta limpieza, podemos utilizar ese archivo Excel para tratar los datos antes de introducirlos en el modelo.

Para ello, se utilizó Python, un lenguaje de scripting que permite realizar un tratamiento rápido y eficiente, debido a las múltiples librerías que lo facilitan.

En cuanto a los datos referentes a la demografía, se tuvieron que tratar algunos de ellos de manera especial antes de volcarlos al modelo, ya que agrupaban varias preguntas. Al modelo se volcaron directamente el género, la edad, la pertenencia a un trabajo esencial y los ingresos netos mensuales, sin embargo, los siguientes han sido tratados:

- Tipo de Hogar: En este caso se juntan dos preguntas, dando lugar a las siguientes posibilidades de la variable.
 - Pareja con hijos.

- Pareja con hijos menores.
- Hogar unipersonal.
- Otro tipo de hogar
- Tipo de Vivienda: En este caso he juntado otras dos preguntas, dando lugar a las siguientes posibilidades, que también engloban los grupos diferenciales:
 - Vivienda con jardín +80m2.
 - Vivienda con jardín -80m2.
 - Vivienda sin jardín +80m2.
 - Vivienda sin jardín -80m2.

En este caso, la diferencia es tener o no jardín, así como el tamaño de la vivienda, que marcamos en 80m2, un umbral que supone una gran diferencia en cuanto a la dificultad de permanecer en el hogar.

- Estado Laboral: En el caso de esta variable hemos tratado la misma, dando lugar a diferentes posibilidades:
 - Desempleado.
 - Inactivo.
 - Activo.

Por otro lado, tenemos las variables que van a transformarse a las características de las necesidades de los humats. Para esto, hemos transformado el grado de dificultad y el grado de aceptación a una escala que pueda usar el modelo. Como se comentó en el apartado anterior, muchas de las preguntas pertenecen a estos dos grupos, que se tratarán por separado.

- Aquellas preguntas que hacen referencia al grado de dificultad, tienen un resultado de 1 a 5. Este resultado se procesará y se escalará de -1 a 1. Como el 1 del cuestionario es la menor dificultad y el 5 es la mayor dificultad, creemos que debería traducirse en: 1 en el cuestionario equivale a 1 en el modelo, siendo el mayor valor que toma la necesidad, mientras que 5 en el cuestionario equivale a un -1 en el modelo, siendo el menor valor que toma la necesidad. El resto de valores se ajustan entre estos dos.
- Aquellas preguntas que hacen referencia al grado de aceptación con determinadas afirmaciones, también tienen un resultado de 1 a 5. Este resultado se procesará y se escalará de -1 a 1. Como el 1 del cuestionario es el menor grado de aceptación y el 5 el mayor, la traducción a valores del modelo será al revés. El 1 del cuestionario equivale a un -1, el menor valor que puede tomar la necesidad, dado que un pequeño grado de aceptación

equivale a un menor grado de satisfacción. El 5 del cuestionario equivale a un 1, el mayor valor que puede tomar la necesidad, dado que un gran grado de aceptación equivale a un mayor grado de satisfacción. El resto de valores se ajustan en el rango del modelo.

Una vez tratados los datos, cada cuestionario se volcará en un agente del siguiente modo:

- El proceso de asignación de datos a las variables se realizará en el proceso de configuración, justo antes de iniciar la simulación.
- Las variables demográficas formarán parte de las características internas del agente.
- Se asignarán los ítem (datos) correspondientes a las diferentes variables del agente: importancias (valores, experiencial y social) y satisfacciones (experiencial, valores y de relaciones sociales).

Resumen de funcionalidades y características

EN este capítulo se recoge un resumen final, explicando y recordando todas las funcionalidades y características finales del proyecto.

En primer lugar se ha realizado un proceso de **recogida de datos**, para lo cual se ha realizado una encuesta y se ha distribuido mediante diferentes medios electrónicos y redes sociales. El objetivo de la misma fue recaudar información y datos acerca de opiniones de diferentes personas sobre las diferentes situaciones que se pueden dar ante la expansión de un virus como el SARS-COV-2. El cuestionario, preparado con la colaboración de una psicóloga, recogía todo tipo de variables que se utilizan en el modelo del HUMAT, así como las variables demográficas que servirán (como trabajo futuro) para clasificar la salida del sistema. Las preguntas del cuestionario pueden separarse en diversos grupos:

- Situación vital (género, tipo de hogar, edad, estado laboral, etc).
- Grado de acuerdo ante la eficiencia de determinadas medidas durante el confinamiento.
- Grado de acuerdo ante la eficiencia de determinadas medidas después del confinamiento.
- Grado de dificultad para cumplir determinadas medidas de contención del virus.
- Preguntas sobre experiencias psicológicas y emocionales.
- Preguntas sobre experiencias sobre los sentimientos.
- Preguntas referentes a las relaciones sociales antes y después de determinadas medidas de contención.

Una vez recogida la información de las encuestas, es necesario limpiar los datos para eliminar valores incompletos o incongruentes. Además, algunas preguntas del cuestionario reflejan el mismo concepto. Esta es una práctica habitual en las ciencias sociales para verificar las respuestas del individuo. Las preguntas que hacen referencia a un mismo concepto se han agrupado como parte de la limpieza.

Los datos obtenidos en esta encuesta se han volcado uno a uno al sistema. Es decir, cada respuesta de la misma ha servido para simular un individuo en el modelo, lo que ha supuesto un total de 2035 individuos en las simulaciones.

En definitiva, los datos obtenidos de la encuesta nos han servido para simular el comportamiento de los agentes y la comunicación entre los mismos del modo más real posible.

Cada agente del modelo tiene una red social que representa su círculo de contactos (vecinos, amigos, familia...) que será utilizada para simular contactos, desde el punto de vista vírico, y para la difusión de información a través de la arquitectura HUMAT. La red social se define como una red *Small Worlds* con las siguientes características:

- Cada agente será relacionado con los cuatro agentes más cercanos a él.
- Existe una probabilidad de reconexión. Si se da el caso, se elimina una de las conexiones existentes en el agente. En lugar de crear esta conexión con un individuo cercano, se creará con uno completamente aleatorio, lo que aporta más realismo al modelo (distancia, nuevas amistades, etc).

Por otro lado, se ha implementado un modelo epidemiológico para simular la expansión del virus en la población. Para esta tarea hemos decidido implementar el modelo SIR (explicado en el Capítulo 5) pero con algunas características diferenciales del modelo estándar, a destacar:

- Se añade un estado más al modelo SIR, el estado de defunción, por lo que los agentes implementados en este modelo pueden pertenecer a más estados y transiciones que en el modelo epidemiológico original.
- Este nuevo estado, además, modifica uno de los anteriores del SIR. Ya no existe el estado "Removed", pasa a ser "Inmune" e indicará inmunidad al contagio, aunque el individuo mantenga los contactos.
- Se elimina la propiedad de Mezclado Homogéneo (todos los individuos tendrán la misma probabilidad de entablar contacto con otro individuo aleatorio del sistema) del modelo SIR, y se busca una aproximación diferente: existirá una mayor probabilidad de entrar en contacto con individuos de su red social, aunque no se elimina por completo la posibilidad de contacto aleatorio, para simular las interacciones callejeras de un individuo.

- Se define contacto como la estancia durante más de 10 minutos con el mismo individuo e incumpliendo la distancia de seguridad de metro y medio.
- Se añaden "medidas de contención" que supondrán diferentes situaciones de expansión del virus, así como el modo de restringir los contactos que tienen, estas medidas son:
 - Sin medida: En este caso la expansión se realiza de la forma estándar, sin limitaciones y, como se comentaba, con los agentes pertenecientes a la red social del individuo tratados con una mayor probabilidad de contacto.
 - Confinamiento completo: En este caso, los individuos que cumplan la medida de contención no tendrán peligro de contagio, ya que es una medida extrema en la que se restringen los contactos. Sin embargo, los individuos que no cumplen la medida mantendrán los contactos con otros que tampoco la cumplen.
 - Confinamiento parcial: Este caso es similar al anterior, sin embargo, los individuos que cumplan la medida siguen teniendo una pequeña posibilidad de mantener contacto.

Para enriquecer el modelo, se utiliza un algoritmo para la difusión de información que permita representar la influencia de la red social de un individuo en su decisión de acatar o no las medidas propuestas por el gobierno. Esta influencia se representa a través de la arquitectura HUMAT, explicada en el Capítulo 4, de nuevo, con algunas variaciones y nuevas características sobre la arquitectura original:

- Se añade un nuevo grupo de necesidades, denominadas necesidades de relación social, que determina la importancia de la socialización para cada individuo, o dicho de otro modo, la dificultad para permanecer aislado.
- El peso de confianza que se le da a las relaciones sociales pasa a ser global para la red social, es decir, cada individuo tendrá un valor de confianza para su red de conexiones, no para cada agente con el que se relaciona.

Pruebas y resultados

LA fase final del proyecto es la etapa de experimentación, donde se probarán y analizarán tres escenarios diferentes: En el primer escenario no se utilizará ninguna medida de restricción de movilidad para los ciudadanos, en el segundo escenario se decretará el confinamiento total, reduciendo drásticamente los contactos entre los HUMATS y en el tercer escenario se probará una medida de confinamiento parcial.

9.1 Primera prueba

En primer lugar, se realizará una prueba en un escenario inicial (comportamiento temprano) del virus, es decir, simulando la expansión en la población sin utilizar ningún tipo de medida de contención. En este caso, entonces, no tiene sentido utilizar el valor de aceptación de la medida, ya que no existe ninguna medida. Lo primero es ajustar los valores de configuración a algo que simule la realidad lo máximo posible. Para ello, se define una población de 2035 agentes, como la que se puede ver en la figura 9.1

- Número inicial de infectados: 50.
- Número de contactos diarios: 2. Damos por supuesto que este número es la media en términos generales y bajo medidas mínimas, tener contacto con dos personas fuera del ámbito de convivencia.
- Probabilidad de contagio: 5% determinada por la media que se ha podido recaudar de datos en internet, para contactos de más de 10 minutos pero menores de 30.
- Tiempo en el que se comprueba si un individuo sigue infectado: 10 días.
- Probabilidad de inmunidad: 10%.

- Probabilidad de mortalidad: 5%. Según fuentes en internet, ronda este porcentaje en lo que se lleva de pandemia.

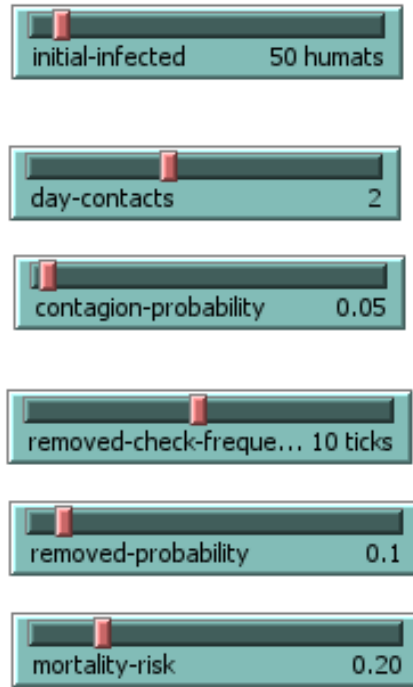


Figura 9.1: Configuración para la primera prueba

Si no se seleccionan medidas de contención, obtenemos los resultados de la figura 9.2, en los que puede verse que la mayor parte de los individuos se han visto contagiados, si es que no todos. Mueren 1023 individuos de los 2035 iniciales, suponiendo la mitad casi exacta de la población.

Para esta primera simulación, vemos que pasan en total de 290 días, en el que el número de muertos sube hasta 1023 y el de inmunes a 430. Como se puede ver en estos datos, se asume que al final de esos días, la gran mayoría (si no toda) de la población simulada se ha visto infectada al menos una vez, con lo que los datos reflejan, de algún modo, la teoría de inmunidad de grupo. Esta simulación acaba cuando no hay más individuos infectados, igual que el resto de las que se han realizado.

9.2 Segunda prueba

Esta segunda prueba consiste en ejecutar el modelo con la misma configuración pero para el caso de confinamiento total. Es importante aquí destacar las diferencias entre el estado inicial y el estado final de la ejecución en lo que se refiere al número de agentes que cambiarán de

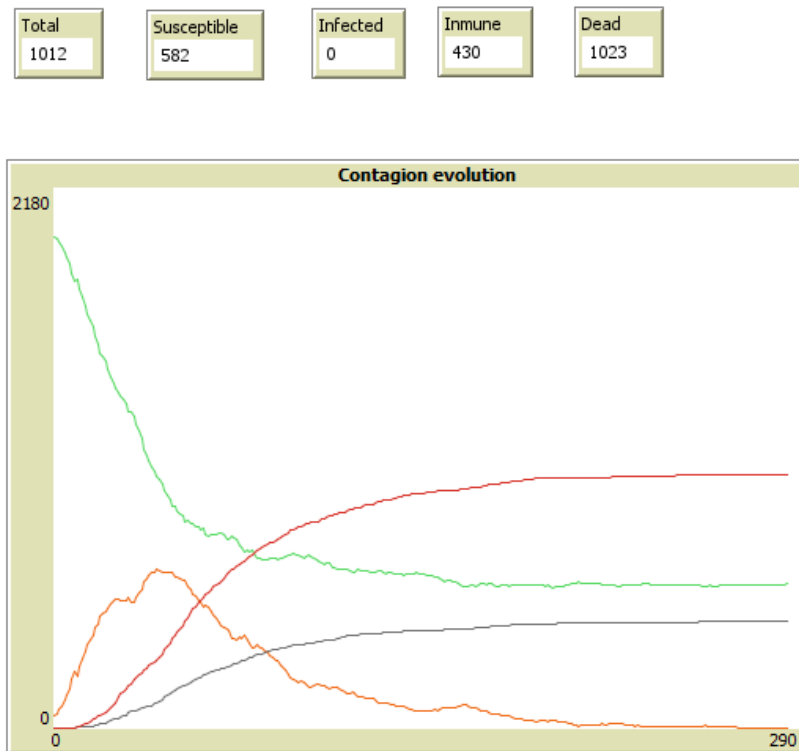


Figura 9.2: Resultados para la primera prueba con medida de contención nula. En verde la línea de individuos susceptibles a lo largo de la simulación, en naranja el número de infectados, en rojo el número de muertos y en gris el número de inmunes.

parecer y acabarán aceptando la medida. El estado previo es similar al de la ejecución anterior, salvo que ahora se le indica al sistema que se aplicará esta nueva medida de contención, lo que genera cierto número de individuos que aceptan la medida. En esta simulación son 333, lo cual, para una población de 2035 individuos es un número muy bajo, en consonancia con lo que se comentó anteriormente acerca de que la adherencia hacia las medidas de contención suele ser baja.

Además, en el resultado de la figura 9.3, se puede observar como tanto la curva de infectados como la de muertos alcanzan valores considerablemente más bajos que en el caso anterior, ya que aunque la aceptación sea baja, el hecho de restringir un gran número de contactos, acaba reduciendo considerablemente las infecciones. Ahora bien, al final de la simulación puede verse que un buen número de individuos han evolucionado su comportamiento y han pasado a aceptar la medida.

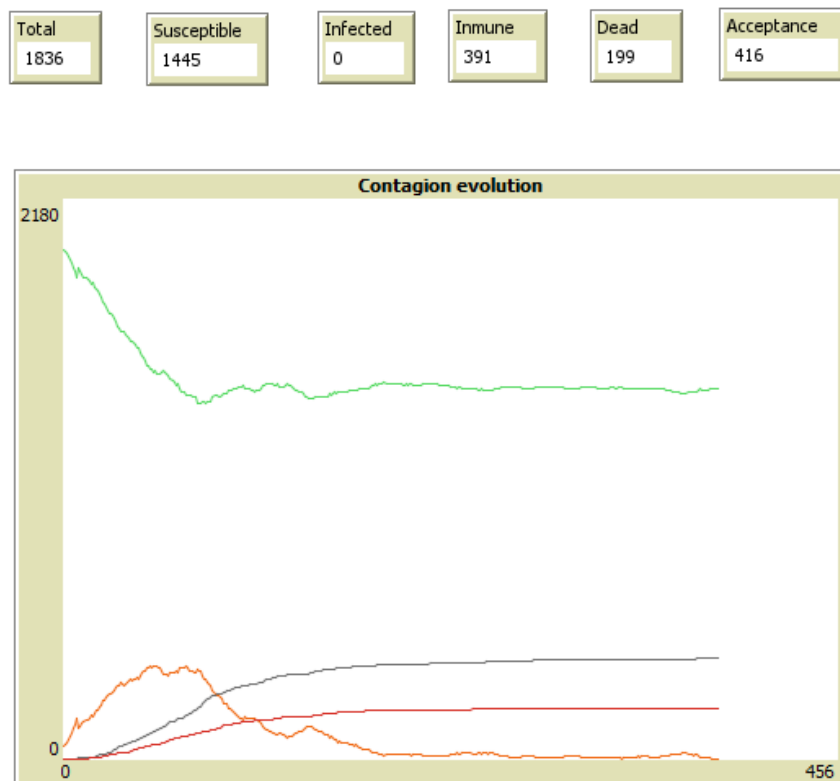


Figura 9.3: Prueba 2: Final de la simulación. En verde la línea de individuos susceptibles a lo largo de la simulación, en naranja el número de infectados, en rojo el número de muertos y en gris el número de inmunes.

9.3 Tercera prueba

Para la última prueba, la medida de contención será el "confinamiento parcial". Como se comentó en la fase de desarrollo, no tiene una gran diferencia con la de confinamiento completo, pero sin embargo producirá más contactos y, por ende, más infecciones. Del mismo modo que en el caso anterior, es importante determinar el estado inicial, que en este caso agrupa a 646 individuos que aceptan la medida de contención. El estado final tras la simulación, se muestra en la figura 9.4. Del mismo modo que en el caso del confinamiento completo, la simulación dura más que en el caso de no aplicar medidas de contención, en concreto 456 días, esto viene dado porque el número de inmunes en el caso de no aplicar medidas crece muy rápido, y prácticamente toda la población se ha infectado al menos una vez. Sin embargo, en este caso de confinamiento parcial, vemos que la curva de infectados y muertos en el modelo resulta menor que en el confinamiento completo. Este resultado puede explicarse por el nivel de adherencia de los ciudadanos a las medidas: la aceptación de la medida es mucho mayor para este último caso (el doble aproximadamente que en el caso anterior antes de comenzar la simulación, que eran 333), y, por lo tanto, se producen menos contactos.

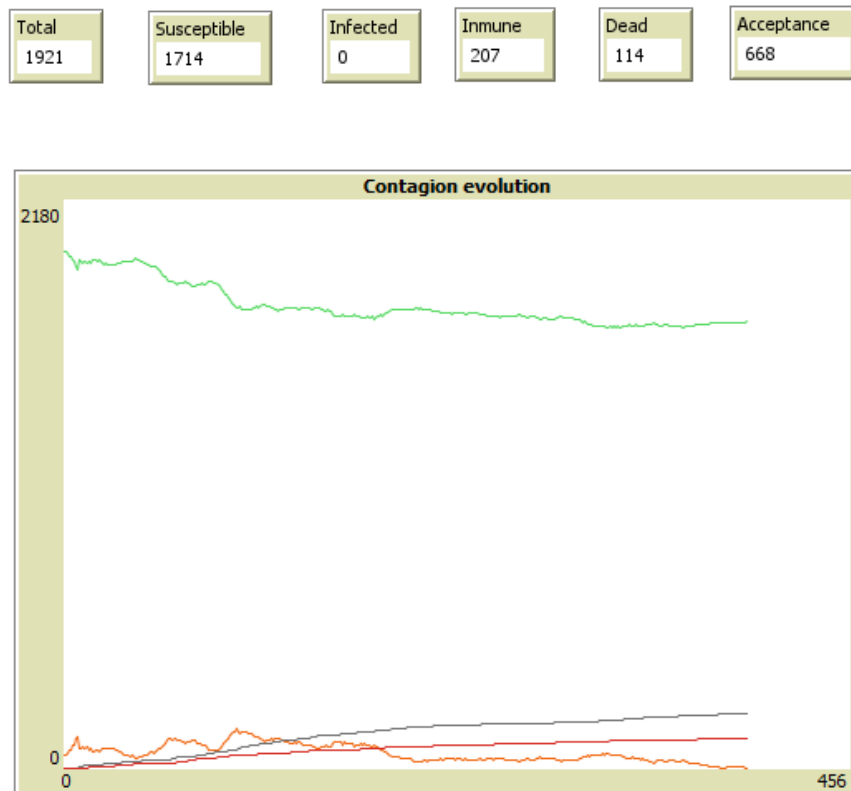


Figura 9.4: Prueba 3: Final de la simulación. En verde la línea de individuos susceptibles a lo largo de la simulación, en naranja el número de infectados, en rojo el número de muertos y en gris el número de inmunes.

Conclusiones y trabajo futuro

EN este capítulo se exponen las conclusiones obtenidas tras finalizar el proyecto, así como diferentes mejoras y avances que se podrían tener en cuenta a trabajo futuro.

El objetivo del proyecto era construir una población de individuos en la que se pudiesen simular diferentes expansiones de un virus epidemiológicos con diferentes características y bajo diferentes medidas de contención. Además, los agentes debían de poseer determinada inteligencia, capacidad de toma de decisiones y capacidad de comunicación.

Se ha logrado un entorno de simulaciones bastante cercanas a la realidad, gracias a las extensiones de los modelos epidemiológicos conocidos, a la herramienta NetLogo, a los datos obtenidos mediante cuestionarios y, sobre todo, a la arquitectura HUMAT (junto a la pequeña expansión que se añadió en este proyecto mediante el nuevo grupo de necesidades).

Cuando una gran cantidad de personas no están dispuestas a cumplir las medidas de contención se observan crecimientos en contagios que tanto en el mundo real como en las simulaciones. Durante las pruebas se ha probado a forzar una mayor aceptación de las medidas, en concreto, del confinamiento parcial, y los casos de contagio y defunción disminuían drásticamente.

También es importante destacar la diferencia que hay entre no aplicar ninguna medida y sí hacerlo. Los casos de defunción bajan enormemente, pero aumenta el malestar general. Aunque parezca evidente, todavía no está claro cuál de las medidas puede ser la mejor para maximizar ambos puntos.

En cuanto al trabajo futuro, se identifican varias líneas de mejora a medio plazo:

- Añadir nuevas medidas de contención del virus, y parametrizar las existentes.

-
- Añadir otros modelos epidemiológicos algo más complejos, que le den más profundidad al sistema.
 - Mejoras en la red de conexiones, mejorando así la red social.
 - Uso de las variables demográficas de los agentes para realizar análisis de resultados de las simulaciones, así como agrupación de los resultados por categorías.

Bibliografía

- [1] P. Antosz, W. Jager, G. Polhill, D. Salt, A. Alonso-Betanzos, N. Sánchez-Marroño, B. Guijarro-Berdiñas, and A. Rodríguez, “Smartees, simulation model implementing different relevant layers of social innovation, human choice behaviour and habitual structures.”
- [2] N. Kasabov, *Introduction: Hybrid intelligent adaptive systems*. International Journal of Intelligent Systems. 6: 453–454, 1998.
- [3] G. Weiss, *Multiagent systems*. Cambridge, MA: MIT Press. ISBN 978-0-262-01889-0, 2013.
- [4] S. F. Railsback and V. Grimm, *Agent-Based and Individual-Based Modeling*, 2nd ed. Princeton University Press.
- [5] A. Quezada and E. Canessa, *Agent-based modeling: A tool for complementing the analysis of social phenomena*.
- [6] M. Martcheva, *Introduction to Epidemic Modeling*.
- [7] M. Newman, *Networks: an introduction*. Oxford, 2010.
- [8] “Netlogo.” [En línea]. Disponible en: <https://ccl.northwestern.edu/netlogo/>
- [9] “Uri wilensky.” [En línea]. Disponible en: <https://ccl.northwestern.edu/Uri.shtml>
- [10] “Github.” [En línea]. Disponible en: <https://github.com/ManuSeijas/tfg-manuel-seijas-carpente>
- [11] “Watts-strogatz model.” [En línea]. Disponible en: https://en.wikipedia.org/wiki/Watts-Strogatz_model

