Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA

Grado en Ingeniería Informática de Servicios y Aplicaciones

Desarrollo de un sistema de generación de series temporales para propósitos de aprendizaje automático

---

Alumno: Patricia Baz Domínguez

Tutores: Miguel Ángel Martínez Prieto
Jorge Silvestre Vilches

Fecha: 16 de septiembre de 2022

A mis padres,
por ayudarme y apoyarme siempre,
anteponiendo mi felicidad a la suya propia.
Gracias.

Lo que no se define no se puede medir.
Lo que no se mide, no se puede mejorar.
Lo que no se mejora, se degrada siempre.
William Thomson Kelvin, físico y matemático británico del siglo XIX.

Para algunos era nada,
para otros quizás el mito
y, al final, la verdad está en el vértigo de lo infinito.
"Belfast", Juancho Marqués

Agradecimientos

A pesar de que un Trabajo de Fin de Grado (TFG) va firmado por una única persona, son muchas las que ayudan a que salga adelante. Cada una de las personas que nombro a continuación me ha ayudado a su manera, aportando lo que estaba a su alcance en cada momento; han sido mi motor en la realización de este trabajo, tanto a nivel académico como psicológico.

En primer lugar, me gustaría dar las gracias a mis tutores, D. Miguel Ángel Martínez Prieto y D. Jorge Silvestre Vilches, así como al profesor D. Aníbal Bregón Bregón, por aceptarme en su equipo UVAGILE, por su infatigable carácter a la hora de buscar un tema que se adaptara a mis gustos y por su apoyo incondicional durante todo el proyecto; gracias por confiar en mí incluso cuando yo no lo hacía. En esta línea, todos los miembros de la datAicademy han contribuído en la consecución de este trabajo.

Quiero agradecer también al resto de personal de la Universidad de Valladolid por transmitir su conocimiento y valores desde la cercanía y la generosidad.

Además, quiero mencionar a mi familia y a mis amigos del pueblo, del grado y del programa SICUE, que siempre tratan de sacar lo mejor de mí y me aconsejan en todas y cada una de las decisiones que he de tomar en busca de mi felicidad.

Resumen

Existe una gran demanda de datos para el entrenamiento de modelos de aprendizaje automático. Sin embargo, la oferta no suele suplir tal necesidad debido a problemas de suficiencia, sesgo, privacidad y coste, entre otros. Los datos sintéticos, datos obtenidos artificialmente tratando de simular datos reales, satisfacen dicho requisito.

En este proyecto se estudian los distintos tipos de datos sintéticos y técnicas para generarlos que existen en la actualidad. Además, se desarrolla una aplicación *web* basada en un sistema *software* capaz de generar datos sintéticos tabulares acerca del estado en el que se encuentra un individuo durante la pandemia que consterna a la población mundial desde el año 2020, el COVID-19. Se obtienen series temporales sobre el COVID-19 mediante una técnica híbrida que combina dos de los grandes métodos de generación de datos sintéticos; a saber, generación basada en reglas y procesos estocásticos. Esta herramienta es adaptable a otros casos de uso simplemente cambiando el modelo matemático que modela el fenómeno que se estudia.

Palabras claves: Aprendizaje Automático, Datos Sintéticos, COVID-19, Series Temporales, Modelo Matemático.

Abstract

There is a huge data demand for training machine learning models. However, the offer does not usually meet this need due to problems of sufficiency, bias, privacy and cost, among others. Synthetic data, artificially obtained data trying to simulate real data, satisfies this requirement.

This project studies the different types of synthetic data and techniques to generate them that currently exist. In addition, a web application based on a software system capable of generating tabular synthetic data about the state in which a person is during the pandemic that has dismayed the world population since 2020, COVID-19, is developed. Time series on COVID-19 are obtained using a hybrid technique that combines two of the major synthetic data generation methods; namely, rule-based generation and stochastic processes. This tool is adaptable to other use cases simply by changing the mathematical model that models the phenomenon under study.

Keywords: Machine Learning, Synthetic Data, COVID-19, Time Series, Mathematical Model.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Los datos y los algoritmos están cambiando las reglas del juego. La Inteligencia Artificial o (*Artificial Intelligence*) (IA) supone un cambio de paradigma en la forma de entender el mundo que supera a la capacidad humana en muchos aspectos. El reconocimiento automático de voz, imagen, vídeo y texto son ejemplos claros. La implantación de esta tecnología es ya una realidad y existe una gran demanda de profesionales en este campo.

Tal es la importancia de estos activos en la sociedad actual que las organizaciones están evolucionando desde una postura clásica para la toma de decisiones, basada en la experiencia, a una toma de decisiones informada, basada en los datos. Hay muchas fuentes, internas y externas, desde las cuales los datos fluyen hacia una organización y ésta debe organizarlos y segregarlos. Un análisis adecuado permite aprovechar el poder de los datos para llegar a información valiosa y procesable, hacer predicciones apropiadas y llevar a cabo una toma de decisiones eficiente. Por lo tanto, estas instituciones deben confiar en lo que aprenden de los datos analizados y evitar recurrir a la toma de decisiones intuitiva [1].

Hay muchas y muy diversas fuentes de información. En particular, cada persona genera grandes cantidades de datos en su día a día; por ejemplo, cuando utiliza sus redes sociales, rellena una encuesta o acude al gimnasio. Estos son recopilados por diversas organizaciones que, a su vez, se encargan de hacer estudios muy variados a partir de ellos con el fin de tomar mejores decisiones. Sin embargo, existen leyes, como la Ley Orgánica de Protección de Datos (LOPD) en España [2], que regulan el uso de estos datos, pues algunos de ellos contienen información personal y, por ende, intransferible. Cuando las organizaciones no pueden usar o compartir este tipo de información, ya sea públicamente o con otros sectores internos de la propia organización, han de generar *pseudodatos*. Es entonces cuando pasa a un primer plano el concepto de dato sintético.

Por tanto, los datos son una materia prima que debe ser pulida para poder ser utilizada; los datos se recogen, se analizan, se agrupan... y, cuando todo esto resulta insuficiente, se generan sintéticamente. Esta carencia se suele producir debido a que toda esta información generada a diario está incompleta, es insuficiente o no es de calidad. Como su propio nombre indica, los datos sintéticos se originan artificialmente y tienen como objetivo simular datos reales. Más concretamente, los datos sintéticos son datos generados algorítmicamente aproximando unos datos originales, pudiéndose utilizar ambos para el mismo propósito [3].

El uso de datos sintéticos ha ganado popularidad recientemente por su demanda para propósitos de Aprendizaje Automático o *Machine Learning* (ML). De hecho, el medio digital *Business Insider* [4] los define como datos generados de forma artificial que pueden ser utilizados para

entrenar modelos de IA (y, por tanto, de ML), en el caso de que los datos reales carezcan de calidad o no sean demasiados [5]. Esta es la explicación principal de la última parte del título de este TFG, es decir, el ML es el cliente de los datos sintéticos que vamos a generar. El aprendizaje automático se centra en construir sistemas computacionales que a partir de datos y percepciones mejoran su rendimiento en la realización de una determinada tarea (sin haber sido explícitamente programados para dicha tarea) [6]. Pero, para conseguir mejorar ese rendimiento, son necesarios conjuntos de entrenamiento adecuados que no siempre se encuentran disponibles.

Los datos sintéticos, aunque en auge actualmente, no son nuevos; tienen un recorrido cuyo inicio se remonta un siglo en el tiempo. Así, resulta oportuno comentar que el modelado científico de sistemas físicos, el cual permite ejecutar simulaciones en las que se puede estimar, calcular o generar puntos de datos que no se han observado en la realidad, tiene una larga historia que se desarrolla simultáneamente a la historia de la física. Por ejemplo, la investigación sobre la síntesis de audio y voz, impulsada por los desarrollos del teléfono y la grabación de audio, se remonta a la década de 1930. Más adelante, a partir de la década de 1970, la digitalización dio lugar a los sintetizadores de *software* [7].

En el contexto del análisis estadístico que preserva la privacidad, en 1993, Rubin [8] creó la idea de datos originales totalmente sintéticos, diseñados para sintetizar, de forma corta, las respuestas de formato largo del Censo Decenal para los hogares. Luego publicó muestras que no incluían ningún registro real de forma larga; en esto preservó el anonimato de las casas [9]. Ese mismo año, la idea de los datos originales parcialmente sintéticos fue creada por Little, quien la usó para sintetizar los valores sensibles en el archivo de uso público [10]. Posteriormente, se planteó una solución sobre cómo tratar los datos parcialmente sintéticos con los datos faltantes.

Para demostrar la importancia y el recorrido de este tipo de datos de una manera más concreta, se repasan, a continuación, varios ejemplos de aplicaciones de los datos sintéticos en la vida real [3]:

- Los datos sintéticos tienen aplicación en el campo del Procesamiento del Lenguaje Natural o *Natural Language Processing* (PLN). El equipo de IA *Alexa* de Amazon [11], por ejemplo, utiliza datos sintéticos para completar los datos de entrenamiento de su sistema de Comprensión del Lenguaje Natural o *Natural Language Understanding* (NLU). Esto les proporciona una base sólida para entrenar nuevos idiomas sin datos de interacción con el cliente existentes, o suficientes.

- Cuando se trata de datos sintéticos, un uso popular para ellos es el entrenamiento de algoritmos de visión. Durante más de un año, el equipo de *Waymo* [12] ha estado generando conjuntos de datos de conducción realistas a partir de datos sintéticos. La compañía utiliza estos conjuntos de datos para entrenar sus sistemas de vehículos autónomos. Es una forma eficiente de incluir escenarios más complejos y variados, en lugar de gastar mucho tiempo y recursos para obtener observaciones de escenarios similares. *Waymo* no es la única compañía que confía en datos sintéticos para este caso de uso: GM Cruise, Tesla Autopilot, Argo AI y Aurora también lo son. En la industria minorista, Amazon también implementó técnicas similares para la capacitación de *Just Walk Out*, el sistema que impulsa las tiendas sin cajero de Amazon *Go*. El equipo generó una cantidad considerable y variada de datos sintéticos de comportamiento del cliente para entrenar su sistema de visión por computadora.

- La institución financiera *American Express* [13] ha estado investigando el uso de datos sintéticos tabulares para el análisis predictivo. Su equipo de ciencia de datos ha investigado

cómo generar datos sintéticos estadísticamente precisos a partir de transacciones financieras, para realizar la detección de fraudes. Ya en el campo de los seguros, donde los datos de los clientes son un recurso esencial y sensible, la empresa suiza *La Mobilière* [14] utilizó datos sintéticos para entrenar modelos de predicción de abandono. El equipo de ciencia de datos modeló datos sintéticos tabulares a partir de los datos de clientes de la vida real, entrenando así los modelos de ML sin comprometer su rendimiento o la privacidad de los clientes.

- Uso de datos sintéticos como método de protección de datos. Por ejemplo, en el campo de la salud, el uso de los datos del paciente está extremadamente regulado. Roche [15] validó el uso de datos sintéticos como reemplazo de los datos de los pacientes en la investigación clínica. El laboratorio alemán Charité de IA en medicina también está trabajando en el desarrollo de datos sintéticos para generar datos para la investigación colaborativa y facilitar la progresión de diferentes casos de uso médico.

Como bien se menciona en este último ejemplo, los datos sintéticos tienen cabida dentro del campo médico. Tal es el caso que la colección que se va a generar en este proyecto es relativa a la actual pandemia que ha consternado a la población mundial los últimos dos años, el COVID-19. COVID-19 es el nombre oficial que la Organización Mundial de la Salud (OMS) le dio en febrero de 2020 a la enfermedad infecciosa causada por el nuevo coronavirus, es decir, el virus SARS-CoV-2. Desde finales del año 2019, el planeta se ha visto inmerso en una pandemia provocada por este virus que ha tenido y tiene consecuencias que afectan el curso normal de la vida de todos los ciudadanos: el trabajo, la educación, la cultura, el ocio... pero, sobre todo, la sanidad. Todo ha sido modificado, en mayor o menor medida, a causa de las restricciones que los gobiernos de los países se han visto obligados a imponer. Esta pandemia ha afectado a pequeños empresarios, autónomos y sus trabajadores, que han perdido su empleo y, en muchos casos, sus ingresos. Por ello, ha sido tan importante encontrar soluciones farmacológicas que puedan paliar los efectos del virus. Distintos equipos investigadores intentaron obtener antivirales cuanto antes para inhibir los contagios, mientras a más largo plazo se trataba de encontrar una vacuna efectiva contra el virus. Durante todo este proceso también ha sido de alta prioridad el hecho de estudiar modelos matemáticos que predijesen el comportamiento de la pandemia en lo sucesivo (a corto y largo plazo) para así disponer de los recursos necesarios a tiempo y evitar el mayor número de muertes posible. Estos recursos eran, principalmente, camas en la Unidad de Cuidados Intensivos (UCI) y respiradores, insuficientes en la mayoría de los hospitales sobretodo durante los "picos" de esta pandemia [16].

Este tipo de modelos, dependientes de varios parámetros, han proporcionado datos más fiables que los propios datos recolectados por el Ministerio de Sanidad español, ya que el proceso de recogida de información es muy complicado. Este desajuste de los datos de COVID-19 en España, que también se ha producido en otros países, es responsabilidad compartida entre las comunidades autónomas y el Ministerio. En ocasiones, cada comunidad hacía un recuento de casos y muertes guiados por definiciones distintas. Por ejemplo, en mayo de 2020 se hizo una revisión de las series porque Sanidad había detectado que algunas comunidades informaban como fallecidos por COVID-19 personas que tenían síntomas, pero que no tenían la enfermedad confirmada con Reacción en Cadena de la Polimerasa o *Polymerase Chain Reaction* (PCR) positiva [17]. Otro ejemplo más de la dudosa fiabilidad de los datos reales lo protagonizó Reino Unido con el escándalo vivido la última semana de septiembre de 2020, en pleno rebrote de la pandemia; un

fallo informático en el recuento de contagios hizo que 16000 casos de coronavirus no se notificaran a tiempo para rastrear los contactos [18].

Por todo esto, el sistema *software* producto de este trabajo va a generar datos sintéticos acerca del COVID-19 a partir de uno de estos modelos, en concreto el modelo SEIR. Además, cabe mencionar que el sistema, aunque pueda parecer totalmente específico, se podría adaptar fácilmente a la generación de datos sintéticos referentes a otras materias con el simple hecho de modificar el modelo matemático base que conforma las reglas.

Antes de finalizar esta introducción, conviene entender aquello que se va a generar con el sistema *software*. Cada registro generado se identifica mediante un número cardinal que representa el día en que se obtienen dichos resultados dentro de una vista reducida de la pandemia, esto es, se genera una **serie temporal**. Una serie de este tipo es una colección de observaciones de una variable recogidas secuencialmente en el tiempo. Estas observaciones se suelen recoger en instantes de tiempo equiespaciados. Si los datos se recogen en instantes temporales de forma continua, hay dos formas de discretizarlos. Una es digitalizar la serie, es decir, recoger sólo los valores en instantes de tiempo equiespaciados, y la otra, acumular los valores sobre intervalos de tiempo. La característica fundamental de las series temporales es que las observaciones sucesivas no son independientes entre sí, y el análisis debe llevarse a cabo teniendo en cuenta el orden temporal de las observaciones. Por ello, los métodos estadísticos basados en la independencia de las observaciones no son válidos para el análisis de este tipo de series [19].

1.1. Planteamiento del problema

Actualmente, el deseo de todos los profesionales del mundo de los datos es que estos presenten las características precisas para adaptarse a la perfección al uso que se les quiera dar. Es decir, lo ideal es poseer una cantidad suficiente de datos para conseguir los propósitos predefinidos y que esta adquisición no sea costosa. Pero, en la realidad, llegan datos en bruto que hay que analizar y tratar para su uso posterior. Y, en muchas ocasiones, estos datos son escasos, caros, están sesgados o son confidenciales, de manera que causan aún más problemas a quienes los requieren.

En consecuencia, surge la necesidad de generar datos sintéticos para suplir todas estas carencias por las que estos conjuntos de datos reales se caracterizan.

La propuesta de este proyecto es estudiar el avance actual al que se ha llegado en la generación de datos sintéticos con el fin de identificar técnicas adecuadas para generar datos tabulares totalmente sintéticos para series temporales. Concretamente, se abordará la construcción de una herramienta *software* de generación de este tipo de datos sintéticos que se evaluará en el contexto de la preocupante y actual pandemia COVID-19.

1.2. Objetivos del trabajo

El presente proyecto se enmarca en la problemática asociada a la generación de datos totalmente sintéticos para series temporales. En esta línea de trabajo se identifican varios objetivos:

- **OBJ-01:** Orientar el uso de las técnicas y herramientas existentes para la construcción de modelos de generación de datos sintéticos. Este objetivo se subdivide en los siguientes subobjetivos para facilitar su consecución:

  - **OBJ-01.01:** Explorar la aplicación de trabajos que hayan seguido la dirección de este proyecto en este problema concreto.
  - **OBJ-01.02:** Caracterizar las propiedades más relevantes que deben presentar los datos usados para propósitos de ML.
- **OBJ-02:** Desarrollar y evaluar un sistema *software* capaz de generar datos sintéticos a partir de ciertos parámetros descriptivos de series temporales.

1.2.1. Restricciones

El alcance y la duración del proyecto están limitados por la carga de trabajo que especifica la asignatura TFG, que son 12 Sistema Europeo de Transferencia y Acumulación de Créditos o *European Credit Transfer System* (ECTS). Es por esta restricción que los objetivos descritos, aunque ambiciosos, no ahondan más en el tema propuesto.

También resulta preciso comentar que la evaluación del sistema se encuentra limitada debido al problema de fiabilidad de los datos reales tomados en España y Reino Unido. Además de los problemas comentados, estos datos no son completamente representativos debido a que no toda la población expuesta al virus se ha realizado los *tests* correspondientes o ha comunicado su positividad, en caso de hacérselos en casa. Los datos generados no van a tener una evaluación de la utilidad tan alta como se quisiera, pero esto no significa que no sean representativos de la realidad, ya que los datos reales disponibles para la comparación no reflejan completamente la realidad vivida durante la pandemia. En resumen, los datos generados reflejan la realidad teórica, no la de los datos recogidos por los países, los cuales están sesgados.

1.3. Estructura de la memoria

El presente documento se encuentra dividido en 6 capítulos, que tratan los diversos objetivos planteados en el proyecto, junto con algunas consideraciones ligadas a sus procesos de desarrollo y gestión del proyecto.

1. Capítulo 1: Es el capítulo actual y trata de introducir al lector en el dominio de este proyecto. En él se especifican el problema que se trata de solucionar con el proyecto (sección 1.1), los objetivos del mismo (sección 1.2) y la estructura de la memoria (sección 1.3).

2. Capítulo 2: Ahonda en el proceso de gestión del proyecto, por tanto, incluye la metodología utilizada durante su desarrollo (sección 2.1), su planificación temporal y económica (sección 2.2 y sección 2.3) y el balance o comparación entre lo estimado y la realidad final (sección 2.4).

3. Capítulo 3: Contextualiza el proyecto dentro de los entornos *data-driven*, el ML y los datos sintéticos, repasando así el entorno de negocio (sección 3.1 y sección 3.2). Además, incluye el estado del arte (sección 3.3) del proyecto y otros recursos de interés ligados al tema que le compete (sección 3.4).

4. Capítulo 4: Expone la propuesta del proyecto, desde el modelo matemático en el que está basado el generador de datos sintéticos (sección 4.1), hasta los destalles de implementación (sección 4.4), pasando además por la explicación del proceso de análisis (sección 4.2) y diseño (sección 4.3) *software*. Contiene, además, dos manuales (sección 4.5).

5. Capítulo 5: Muestra las métricas que evalúan el sistema generador de datos sintéticos y el conjunto de datos reales con los que se compara (sección 5.1). Además, explica el proceso de evaluación (sección 5.2) y razona los resultados obtenidos (sección 5.3).

6. Capítulo 6: Analiza el resultado global del proyecto, estudiando el grado de consecución de los objetivos y sacando conclusiones acerca del trabajo y del desempeño personal durante la realización de este TFG (sección 6.1). Termina con un análisis de posibles trabajos futuros (sección 6.2).

Se advierte que, durante toda la memoria, los nombres de archivos y los comandos en Python se mostrarán con una fuente monoespaciada, mientras que los extranjerismos y términos técnicos se escribirán en cursiva.

Capítulo 2

Planificación

En este capítulo se presentan, por un lado, la metodología seguida para alcanzar los objetivos del presente trabajo (sección 2.1) y, por otro lado, la planificación previa del proyecto (sección 2.2 y sección 2.3) junto con un análisis crítico de las desviaciones que se han presentado respecto a ella (sección 2.4).

2.1. Metodología de trabajo

Este TFG se ha realizado siguiendo los principios de UVAGILE [20], una metodología usada exclusivamente, hasta el momento, para la docencia universitaria. Aún está en desarrollo; comenzó a usarse en la Universidad de Valladolid para el desarrollo de asignaturas (proyectos de aprendizaje colectivo) y, actualmente, se está extendiendo para su uso en TFGs (variante para proyectos de aprendizaje individual) o incluso prácticas en empresa.

UVAGILE enfoca los procesos de enseñanza-aprendizaje como proyectos de aprendizaje ágiles, cuya dinámica está basada en el marco de trabajo ágil *Scrum* [21]. Además, persigue garantizar la trazabilidad del aprendizaje mediante entregas iterativas e incrementales. Estas entregas, junto a su evaluación y posterior retroalimentación, permiten mejorar el producto.

A continuación, se describen los roles, los eventos y los artefactos de UVAGILE de forma genérica, para después concretar cómo se han materializado en este proyecto.

2.1.1. Roles

La planificación y ejecución de un proyecto de aprendizaje UVAGILE requiere la participación de varios roles, cuyas características y responsabilidades se plantean a continuación.

Los roles de UVAGILE están inspirados en los correspondientes roles de *Scrum*, aunque adaptados al entorno académico en el que se realiza el TFG.

Estudiante

Inspirado en el rol *developer* (desarrollador) de *Scrum*, el estudiante es el alumno matriculado del TFG, encargado de construir el producto objeto de dicho proyecto y de defenderlo ante el tribunal. Por tanto, se trata del rol que soporta una mayor responsabilidad.

El estudiante asume, entre otras, las siguientes responsabilidades:

- Plantear, planificar cronológicamente y estimar la dificultad de las tareas a realizar durante el *Sprint* para alcanzar el objetivo del *Sprint*, consensuado en la *reunión de inicio*.
- Ejecutar dichas tareas de manera que cumplan todos los criterios de aceptación; y esto dentro del límite temporal prefijado.
- Participar en todos los eventos planificados para el *Sprint*.
- Mantener una comunicación fluida con el profesor (a través de los canales establecidos). También con la comunidad si fuera necesario.
- Tener actualizado y organizado el *espacio de trabajo compartido* en todo momento.
- Tener actualizado el *tablero del proyecto* en todo momento.
- Tener actualizado el *cuaderno de trabajo* en todo momento.

Profesor

El profesor es la persona responsable de plantear un proyecto de calidad para que el estudiante alcance los objetivos de aprendizaje considerados en la guía docente del TFG. También se encarga de orientar al estudiante durante todo el proyecto, con el firme objetivo de que la calidad del producto final sea máxima.

Está inspirado en el rol de *product owner* de *Scrum*, aunque también adopta algunos aspectos propios del *scrum master* al servir al resto del equipo y facilitar la comunicación entre los miembros de este. El rol de profesor puede ser adoptado por una o varias personas, dependiendo de si el TFG tiene uno o varios tutores asignados.

El profesor asume, entre otras, las siguientes responsabilidades:

- Definir el objetivo del *Sprint* y detallar, en el tablero de aprendizaje, las historias de proyecto que se deben abordar para completar cada uno de ellos.
- Asegurar que todos los eventos previstos en la metodología se desarrollan en tiempo y forma.
- Participar en todos los eventos planificados para el *Sprint*.
- Actuar como mediador en el evento de *comunicación de progresos*.
- Ayudar al estudiante a resolver sus bloqueos.
- Fomentar la comunicación a través de los canales establecidos.
- Proporcionar *feedback* valioso al estudiante de forma regular y en plazos de tiempo suficientemente cortos como para que la retroalimentación no pierda su vigencia.

En el caso de este proyecto, este rol es adoptado por dos personas, profesores de la Universidad de Valladolid.

Comunidad

La comunidad es una de las novedades que introduce UVAGILE frente a *Scrum*. Trata de establecer un entorno de aprendizaje colectivo agrupando a otros alumnos que coinciden con el estudiante en la realización de su TFG, tanto temporalmente como en su temática. Además, la comunidad puede albergar otras personas interesadas en aprender y aportar valor al grupo, ya sean alumnos, profesores o expertos en el tema objeto de estudio.

La comunidad asume, entre otras, las siguientes responsabilidades:

- Participar activamente en el evento de *comunicación de progresos*, retroalimentando oralmente al estudiante de forma constructiva.
- Ayudar al estudiante a resolver sus bloqueos.
- Fomentar la comunicación a través de los canales establecidos.

En el caso de este proyecto, este rol es adoptado por 8 personas: la propia estudiante; sus dos tutores; dos alumnas que también desarrollan su TFG en el ámbito de la ciencia de datos y cuyos proyectos comenzaron en la misma fecha y siguen una planificación temporal similar; un tercer tutor común a ellas y experto en el tema de este proyecto; y dos alumnas de un curso inferior, interesadas en formarse en ciencia de datos.

Tribunal

Inspirado en el rol de *cliente* de *Scrum*, propio de cualquier proyecto en el que se construye un producto para un usuario final, el tribunal es quien adquiere la responsabilidad de valorar crítica y objetivamente el resultado final del proyecto. Este rol lo desempeñan tres profesores de la titulación, entre los que no se puede encontrar ninguno de los tutores del estudiante.

De acuerdo con el *Reglamento específico relativo a la elaboración y evaluación del Trabajo Fin de Grado en la Escuela de Ingeniería Informática de Segovia de la Universidad de Valladolid*, la comunidad asume, entre otras, las siguientes responsabilidades:

- Evaluar el producto de aprendizaje construido por el estudiante de forma objetiva y atendiendo a los objetivos de aprendizaje del TFG.
- Evaluar objetivamente la calidad de la documentación técnica generada como parte del producto de aprendizaje construido por el estudiante.
- Evaluar objetivamente la calidad de la defensa del TFG realizada por el estudiante.

2.1.2. Eventos

La dinámica de un proyecto de aprendizaje UVAGILE requiere la convocatoria de una serie de eventos para establecer un proceso de trabajo transparente y basado en la comunicación frecuente y regular entre todos los roles participantes en el proyecto. Estos eventos tienen como objetivo principal evitar bloqueos por parte del estudiante.

Los eventos de UVAGILE, listados y descritos a continuación, están inspirados en los correspondientes eventos de *Scrum*, aunque adaptados al entorno académico en el que se realiza el TFG.

Sprint

Es el contenedor del resto de eventos y el cual estructura el proyecto, que se divide en *Sprints*, todos con la misma duración. Esta división permite planificar el trabajo a corto plazo; priorizar aquellos aspectos del proyecto que proporcionen un mayor valor en cada momento; revisar frecuentemente los avances del proyecto y alinearlos respecto a los objetivos esperados, en base a la generación de *feedback* por parte del profesor y la comunidad; y llevar a cabo un ritmo de trabajo sostenido en el tiempo y acotado de acuerdo con las restricciones que impone la asignatura "Trabajo Fin de Grado".

Al comienzo de cada *Sprint*, se determina el objetivo de este, que comprende, además de las mejoras, todas aquellas historias de proyecto que han de abordarse en el corto plazo. Cada historia de proyecto se enmarca en un determinado objetivo y se caracteriza por un conjunto de resultados esperados o criterios de aceptación, los cuales son alcanzados mediante la realización de diferentes tareas.

Gracias a la celebración de ceremonias y a la realización de las tareas planificadas, los *Sprints* son eventos dinámicos. El *Sprint* comienza con una descripción de su alcance, que incluye una selección de las historias de proyecto más prioritarias de acuerdo con el estado del proyecto hasta el *Sprint* anterior. El profesor proporciona el alcance del *Sprint* al estudiante a través del tablero de proyecto.

El *Sprint* siempre comienza el mismo día de la semana (en el caso de este proyecto, el miércoles) y las ceremonias también se celebran siempre el mismo día de la semana (en el caso de este proyecto, el martes), como puede observarse en la Tabla 2.1.

	Lunes	Martes	Miércoles	Jueves	Viernes
Semana 1			INICIO DEL *SPRINT* #1 (basado en el alcance provisional)		
Semana 2		*Reunión de Inicio*			
Semana 3		*Reunión de Sincronización*			
Semana 4		*Reunión de Sincronización*			
Semana 5		*Comunicación de Progresos Retrospectiva*	INICIO DEL *SPRINT* #2 (basado en el alcance provisional)		Generación de *Feedback* (*Sprint* #1)
Semana 6		*Reunión de Inicio* (propuesta de la planificación final)			

Tabla 2.1: Esquema temporal de un *Sprint* UVAGILE

Reunión de inicio

Como su propio nombre indica, da inicio al *Sprint*. En ella, participan el estudiante y el profesor. Al finalizar la *reunión de inicio*, el objetivo del *Sprint* debe haber quedado consolidado y el trabajo planificado. El objetivo del *Sprint* consta de dos componentes: una componente incremental, que incluye las nuevas historias de proyecto a ejecutar durante el *Sprint* y una componente iterativa, con las mejoras derivadas de las correcciones del *Sprint* anterior.

A partir de las historias de trabajo que se ha acordado abordar (propuestas por el profesor y discutidas con el estudiante), el estudiante define las tareas necesarias para satisfacer cada una de estas historias, caracterizándolas por una descripción, una fecha de finalización prevista y una complejidad en puntos de historia [1]. En el caso de este proyecto, los puntos de historia de una tarea tratan de coincidir con el número de criterios de aceptación de la misma, aunque no en todos los casos es así debido a que no es posible tomar cualquier número de puntos de historia si no solo los de la sucesión de *Fibonacci*.

La reunión se realiza tras la entrega del *feedback* del *Sprint* anterior por parte del profesor al estudiante, que coincide con el final de la primera semana del *Sprint*. Se trata de una reunión de 30 minutos de duración en la que el estudiante y el profesor debaten sobre la idoneidad del alcance provisional planteado al inicio del *Sprint* y consolidan el objetivo del *Sprint*, teniendo en cuenta el *feedback* generado para incluir historias de proyecto ya comenzadas pero no finalizadas con éxito.

Tras esta reunión, el estudiante plantea la planificación del *Sprint*, de acuerdo con las tareas que proponga para alcanzar el objetivo de este, en un plazo no superior a 2 días.

Por tanto, los resultados de esta reunión son la consolidación del objetivo del *Sprint* y la planificación del *Sprint* en el *tablero de proyecto*.

Reunión de sincronización

Inspirada en el *Daily* de *Scrum* pero realizada semanalmente en lugar de cada día, tiene como objetivo que el estudiante comunique los avances y bloqueos acontecidos desde la última reunión y el plan de trabajo previsto hasta la siguiente. En esta reunión participan el estudiante y el profesor.

La reunión se realiza semanalmente (el mismo día, a la misma hora y en el mismo sitio , preferiblemente) y tiene una duración máxima de 15 minutos, en los que el estudiante expone sus avances, sus bloqueos y su plan de trabajo sin interrupción del profesor ni debate respecto a ninguno de ellos. Cualquier cuestión que requiera un debate posterior puede realizarse al finalizar la reunión. El resultado de esta reunión es garantizar la transparencia en la marcha del proyecto.

En el caso de este proyecto, se realiza los martes a las 15:30h vía *online* mediante videollamada de *Teams*.

Comunicación de progresos

Inspirada en la *revisión del Sprint* de *Scrum*, tiene como objetivo retroalimentar el desarrollo del proyecto y preparar su defensa, en base al *incremento* de producto consolidado. En esta reunión participan el estudiante, el profesor y la comunidad. La presentación del proyecto expone una descripción de alto nivel, su planificación, la propuesta de solución, los resultados obtenidos

[1] Un punto de historia (*story point*) es una unidad de medida relativa, decidida y utilizada por equipos individuales de *Scrum*, para proporcionar estimaciones relativas del esfuerzo para completar los requisitos [22].

y un análisis crítico del desempeño durante el *Sprint*, que comprende las conclusiones a las que llega el propio estudiante acerca de su desempeño y el trabajo futuro previsto para el siguiente *Sprint*, con el objetivo de proporcionar una visión actualizada del proyecto de acuerdo con el estado en el que se encuentra en cada momento. La *comunicación de progresos* se lleva a cabo dentro de la *Actividad de Seguimiento*, una actividad colectiva en la que también se realizan las comunicaciones de progresos del resto de proyectos en desarrollo por parte del grupo de alumnos que forma parte de la comunidad.

La *comunicación de progresos* se realiza el último día del *Sprint* y tiene una duración máxima de 1 hora por alumno (30 minutos para la presentación y 30 minutos para el debate). El estudiante, realiza la presentación del proyecto de forma oral, simulando en tiempo y forma el acto de defensa del TFG. El profesor hace de facilitador o mediador de un turno de debate que tiene como objetivo valorar la forma y el fondo de la presentación y sugerir mejoras. Finalmente, los miembros de la comunidad participan en el debate, con el objetivo de incorporar otros puntos de vista a la discusión y enriquecer el conocimiento colectivo.

El resultado de la *comunicación de progresos* es una presentación actualizada del proyecto de acuerdo con el incremento de producto consolidado y la retroalimentación recibida durante la reunión.

Retrospectiva

Inspirada en la *retrospectiva* de *Scrum*, tiene como objetivo revisar el proceso de trabajo seguido durante el *Sprint*, para poner en valor los aspectos positivos observados e incorporar mejoras en el mismo, de cara al siguiente *Sprint*. En esta reunión participan el estudiante y el profesor. Es recomendable que la *retrospectiva* se lleve a cabo sobre un tablero que facilite que el alumno pueda expresar los aspectos positivos y negativos observados en el *Sprint*, además de exponer sus sugerencias de mejora. En el caso de este proyecto, se utiliza un tablero *online* (*EasyRetro*) para asegurar así que todos los participantes tienen una visión compartida y en tiempo de real de todos los aspectos que se plantean en la *retrospectiva*.

La *retrospectiva* es la última actividad del *Sprint*, tiene una duración inferior a 30 minutos y se realiza dentro de la *Actividad de Seguimiento*, una vez finalizados los eventos de *comunicación de progresos* correspondientes a los TFGs de los alumnos miembros de la comunidad (la *retrospectiva* es común para todos los TFGs).

Los resultados de la *retrospectiva* se materializan en un análisis crítico, a nivel individual y colectivo, sobre el proceso de trabajo seguido en el *Sprint* actual y un plan de mejora individual (para cada proyecto).

2.1.3. Artefactos

Los artefactos en UVAGILE son subproductos del TFG que, por un lado, soportan la dinámica de trabajo y, por el otro, describen el valor consolidado en base a los avances del estudiante en el proyecto.

Estos artefactos se describen a continuación. Algunos están inspirados en los artefactos correspondientes de *Scrum*, pero otros son propuestas nuevas de esta metodología.

Espacio de trabajo compartido

Se trata del repositorio de trabajo del TFG, en el que el estudiante almacena (de forma organizada en carpetas) todos los subproductos que va generando (código fuente, memoria, figuras, esquemas, diagramas...). Es accesible a todos los participantes en el proyecto, para así garantizar una visión única y actualizada del mismo. Tiene una copia de respaldo en la nube y un sistema de control de versiones para garantizar la trazabilidad de cada uno de los subproductos a lo largo del proyecto.

La jerarquía de directorios de este espacio es la siguiente:

- Documentación: contiene todos los documentos relacionados con la gestión académica del TFG (comunicación de inicio, solicitud de defensa...).
- Incrementos: contiene un subdirectorio por cada *Sprint*, dentro del cual se incluyen todos los subproductos que forman parte de dicho incremento (ver apartado 2.1.3).
- Personal: contiene los resultados temporales que genera el estudiante durante la realización del proyecto; es considerado el directorio de trabajo del estudiante, que tiene libertad para gestionarlo de acuerdo con sus necesidades.
- Recursos: contiene los recursos necesarios para la realización del TFG (colecciones de datos, referencias bibliográficas...) organizados en los subdirectorios necesarios.

El estudiante es el responsable de mantener organizado y actualizado el espacio de trabajo compartido, para que así el profesor sea capaz de acceder a las últimas versiones de cada uno de los productos del proyecto, garantizando la transparencia que tanto se persigue en esta metodología.

Asimismo, este espacio de trabajo debe facilitar la comunicación y colaboración entre estudiante y profesor, además de con los miembros de la comunidad. La interacción entre estudiante y profesor se lleva a cabo mediante un canal privado (en el caso de este proyecto, el título es el nombre del estudiante), cuyas condiciones de uso son pactadas previamente entre ellos. La interacción con el resto de la comunidad se realiza en un canal público (en el caso de este proyecto, el título es "General") cuya dinámica se ajusta a las condiciones establecidas por la propia comunidad.

El despliegue actual se lleva a cabo sobre un equipo *Teams* en el que participan todos los miembros de la comunidad. El equipo UVAGILE tiene un canal "General" abierto a todos los miembros de la comunidad (asociado a ese canal existe un espacio de almacenamiento, gestionado por los profesores, en el que se mantienen recursos de interés colectivo). El equipo UVAGILE tiene un canal propio para cada TFG, en el que participan el estudiante y el profesor. Cada canal tiene un espacio de almacenamiento propio sobre el que se despliega el espacio de trabajo compartido del TFG. El equipo también tiene un canal de "Coordinación" en que el que participan los profesores responsables de algún TFG.

Tablero de proyecto

El *tablero de proyecto* adopta aspectos propios de los tableros de uso habitual en metodologías como *Kanban* o *Scrum*. Es accesible por el profesor y el estudiante, para así garantizar una visión única y actualizada del proyecto. El tablero consta de 6 columnas, destinadas a organizar la especificación de los objetivos del proyecto y las tareas planificadas para alcanzarlos. En el caso

de este proyecto, se ha utilizado la herramienta *Trello* [23]. En la Figura 2.1 puede observarse una instantánea del tablero de este proyecto.

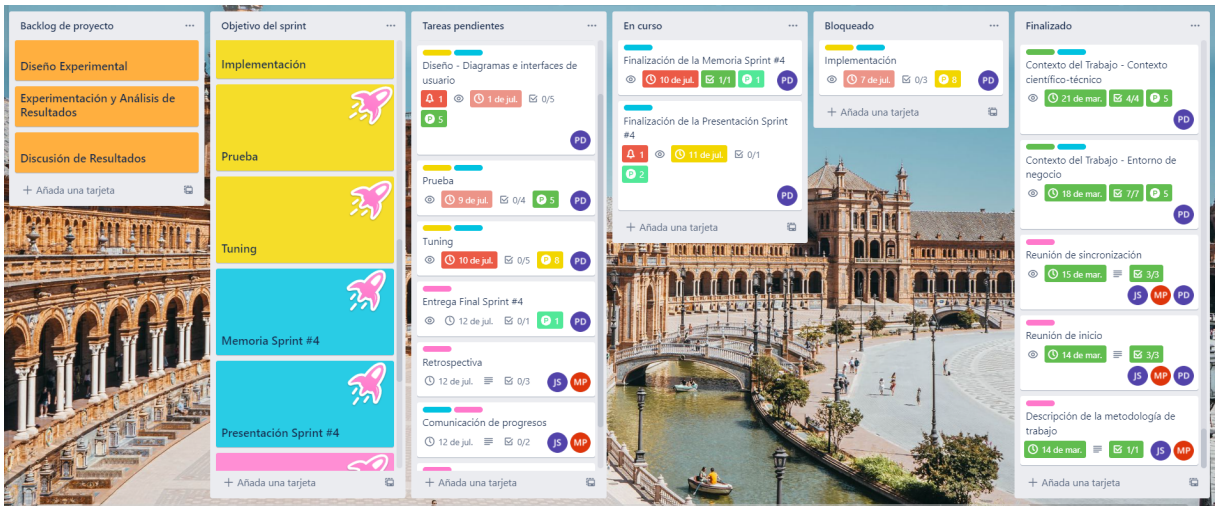


Figura 2.1: *Trello*

Los objetivos se especifican en forma de historias de proyecto, en las que se plantea una breve descripción del cometido de la historia (en el ámbito del proyecto) y una especificación de los resultados esperados a su finalización (que actúan como criterios de aceptación de la historia).

Las tareas plantean una breve descripción del trabajo a realizar en ellas, una estimación del esfuerzo necesario para llevarlas a cabo (en puntos de historia), además de la fecha prevista de finalización y una lista con los resultados esperados. El conjunto de los resultados esperados en todas las tareas asociadas a una historia de proyecto garantiza la obtención de los resultados asociados a la historia que las contiene.

La especificación de los objetivos del proyecto se lleva a cabo en las dos primeras columnas del tablero. La columna "*Backlog* de proyecto" (columna 1) es equivalente al *Product Backlog* de *Scrum* y contiene la especificación de las historias de proyecto cuya ejecución aún no ha sido planificada. Tomando como referencia *Scrum*, el *tablero de proyecto* soporta el *backlog* de producto y el *backlog* de *Sprint*, facilitando la especificación de las historias de proyecto correspondientes. Además, el *tablero de proyecto* facilita la especificación de todas las tareas planificadas para alcanzar el "Objetivo del *Sprint*" (columna 2).

Incremento

Inspirado en el artefacto de *Scrum* con el mismo nombre, es un avance consolidado del producto que permite al estudiante acercarse hacia el objetivo de éste.

El *incremento* de producto materializa el objetivo del *Sprint* y es aditivo a todos los *incrementos* consolidados en los *Sprints* anteriores. En el caso de este proyecto, incluye la memoria del proyecto, la presentación de los avances consolidados y el código fuente, datos, etc. generados durante el *Sprint* y se presenta ante la comunidad en la *comunicación de progresos*, apoyando así el empirismo. El trabajo no se puede considerar parte de un *incremento* a menos que cumpla con la definición de hecho [2].

[2]La definición de hecho (*definition of done*) es una descripción formal del estado del *incremento* cuando cumple

Cuaderno de trabajo

El *cuaderno de trabajo* es una herramienta fundamental en el desarrollo del proyecto, ya que ofrece la posibilidad de tener claro el esfuerzo que se invierte en él, así como las diferentes actividades que se realizan para conseguir los objetivos de aprendizaje planificados *Sprint* a *Sprint*. Por este motivo, el *cuaderno de trabajo* busca introducir una rutina en el ámbito de trabajo del estudiante centrada en el control del tiempo y la (auto)crítica sobre el esfuerzo invertido.

Consiste en un documento en forma de tabla donde el estudiante recoge la fecha, la duración y la descripción de cada uno de los esfuerzos que dedica al TFG. También se especifica el *Sprint* al que pertenece dicha dedicación. La objetividad es necesaria en lo que se refiere a la planificación del trabajo y, por ello, se necesitan datos objetivos que ayuden a aprender de las decisiones tomadas y ser más eficientes en el futuro.

Esta actividad se desarrolla durante todo el proyecto, y tiene una entrega al final de cada *Sprint*.

Retroalimentación

La *retroalimentación* o *feedback* alberga todos los comentarios y críticas constructivas que recibe el *incremento* del estudiante al final de cada *Sprint*. Hay dos fuentes de *feedback*: el profesor y la comunidad, que producen *feedback* sobre distintas parte del *incremento*, mediante distintas vías y en situaciones diferentes. Ambos se complementan de forma que el estudiante es informado de todas las deficiencias identificadas, permitiéndole mejorarlas para el siguiente *Sprint*.

Del profesor El profesor recibe la memoria y el código fuente y los evalúa objetivamente a partir de los criterios de evaluación del TFG y de los criterios de aceptación de las tareas del *Sprint*.

Una vez hecho esto, se lo entrega vía *Teams* al estudiante con los fallos encontrados y sus respectivas correcciones. Estas correcciones puedes ser estrictas, esto es, inapelables, o pueden ser sugerencias de cambio para que el estudiante reflexione sobre ellas y decida qué hacer. También cabe mencionar que se trata de alertas en cuanto a la forma y el contenido del producto.

De la comunidad La comunidad asiste a la presentación oral del *incremento* y solamente puede juzgar la presentación y el contenido que se comenta en ese momento. No es posible valorar la memoria y el código salvo que se expongan porciones de estos en dicha presentación. Es posible intuir, entonces, que la comunicación entre la comunidad en este tipo de *feedback* es oral en su totalidad, lo que requiere de un facilitador: el profesor. Este cede el paso a los participantes según su nivel de experiencia en el asunto. En el caso de este proyecto, uno de los tutores asigna el turno de palabra, en primer lugar, a los alumnos espectadores para seguir con el resto de alumnos de TFG y terminar con el resto de tutores, en concreto, él mismo.

Por una parte, el *feedback* de la comunidad permite escuchar distintos puntos de vista acerca de la parte más estética de la presentación. Por otra, mediante esta evaluación es posible saber si lo que se cuenta es comprensible para todo tipo de oyente (expertas y desconocedoras del tema que se trata). Esta dualidad es muy útil e innovadora.

con las medidas de calidad requeridas para el producto.

2.2. Planificación temporal

La planificación inicial de este proyecto era de 4 *Sprints* con una duración de 4 semanas y una carga de trabajo de 75 horas cada uno:

- Primer *Sprint* (16 mar - 19 abr, con descanso en el periodo 6 abr - 19 abr)
- Segundo *Sprint* (20 abr - 17 may)
- Tercer *Sprint* (18 may - 14 jun)
- Cuarto *Sprint* (15 jun - 12 jul)

A pesar de que todos los *Sprints* han de tener la misma duración, el segundo *Sprint* tiene un desfase de dos semanas debido a las vacaciones de Semana Santa.

La planificación temporal de este trabajo se ha realizado *Sprint* a *Sprint*; los objetivos del proyecto se abordan por *Sprints* en forma de subobjetivos y se va construyendo el objetivo global de forma incremental.

Para planificar todo el trabajo que se debe realizar es esencial analizar las diferentes tareas que conforman el *Sprint* y cuya finalización implica alcanzar los objetivos establecidos al inicio del mismo. Una vez hecho este análisis, se lleva a cabo una estimación del tiempo necesario para realizar cada una de las tareas.

Por este motivo, se definen una serie de tareas descriptivas (análogas a las historias de usuario) y se estima el esfuerzo que hay que dedicar a cada una de ellas utilizando los puntos de historia. Por tanto, la dificultad de cada tarea se expresa en puntos de historia, que denotan el esfuerzo relativo que supone realizar cada tarea frente otras, asignados por la propia alumna.

Las tareas que componen este proyecto por *Sprints* se listan a continuación junto a los subobjetivos de cada *Sprint* y, además, se puede observar la estimación de cada una de ellas en forma de cronograma mediante diagramas de *Gantt*.

En esta sección, dado que se trata la planificación temporal por *Sprints*, se añade la planificación temporal de un quinto *Sprint* que hubo que añadir debido a los retrasos sufridos en los *Sprints* anteriores. La cronología final del proyecto se explica en la subsección 2.4.1.

2.2.1. *Sprint* #1

El *Sprint* #1 lo define un objetivo del *Sprint* que comprende 9 subobjetivos, que se abordarán mediante la realización de 11 tareas.

Subobjetivos Los subobjetivos de este *Sprint* son:

S1.1 Caracterización del Proyecto. Realización de diferentes actividades que más adelante se reflejarán en la memoria, pero que requieren cierto trabajo adicional además de su mera redacción. Estas actividades incluyen describir la motivación del proyecto, definir los objetivos del proyecto, establecer una metodología de trabajo adecuada, plantear una planificación para su consecución, elaborar una introducción descriptiva al proyecto y elaborar unas conclusiones relevantes de acuerdo a los objetivos establecidos y los trabajos realizados.

S1.2 Entorno de negocio. Caracterización de los agentes y factores que tienen cierta influencia sobre las necesidades existentes en la sociedad que se pretenden satisfacer con la propuesta del proyecto.

S1.3 Contexto científico-técnico. Asentamiento de las herramientas y recursos existentes relacionados con la generación de datos sintéticos, con el fin de adquirir el conocimiento necesario y estar en disposición de construir una memoria de proyecto auto-contenida.

S1.4 Estado del arte. Representación de los últimos avances en la investigación aplicada a la generación de datos sintéticos, los cuales pueden proporcionar información de gran valor respecto al enfoque a seguir.

S1.5 Memoria *Sprint* #1. Redacción de una memoria de proyecto que plantea una descripción general del mismo, una explicación detallada de su desarrollo técnico y una exposición crítica de los resultados alcanzados.

S1.6 Presentación *Sprint* #1. Elaboración de una presentación que introduzca y contextualice el proyecto, explique los progresos realizados y concluya con una valoración crítica del estado del proyecto de acuerdo a la planificación y a los objetivos definidos. La presentación es autocontenida y refleja el estado del proyecto en el momento actual.

S1.7 Planificación del *Sprint*. Celebración de un evento de planificación al comienzo del *Sprint* para elegir qué historias de proyecto (del *Backlog* de proyecto) se van a abordar en el *Sprint*, dando lugar al "Objetivo del *Sprint*". Para cada historia individual, se consensúan una serie de "Resultados esperados", que delimitan el alcance del trabajo dedicado a cada una de las historias. Partiendo de estos objetivos y este alcance, se determina el conjunto de tareas concretas, cuya realización genere un incremento que los satisfaga plenamente. Estas tareas se distribuirán de forma adecuada a lo largo del *Sprint* definiendo fechas límite de realización adecuadas. Toda la planificación resultante se reflejará en el tablero de proyecto para asegurar una transparencia y una sincronización adecuadas.

S1.8 Desarrollo del *Sprint*. Comunicación efectiva, durante el transcurso del *Sprint*, través de los recursos disponibles, como las reuniones de sincronización programadas semanalmente, el tablero de proyecto, y las reuniones de trabajo extraordinarias para abordar aspectos concretos del alcance del *Sprint* (en caso de ser necesarias).

S1.9 Finalización del *Sprint*. Consolidación del incremento del producto al final del *Sprint*. En el contexto de este proyecto, esto implica depositar todos los resultados generados (memoria, código fuente y otros) en la carpeta del incremento correspondiente, en el espacio de trabajo; realizar una presentación que describa el estado actual del proyecto y los avances realizados, con el fin de poner en común con el resto de participantes el progreso en los objetivos del proyecto; y llevar a cabo una reflexión crítica sobre el proceso de trabajo, con el fin de asegurar una mejora continua a lo largo del proyecto. A partir de estos resultados, los tutores generan *feedback* adecuado tanto sobre el incremento entregado como sobre la dinámica de trabajo.

Tareas A continuación, se listan las tareas de este *Sprint*, a excepción de las tareas colectivas, esto es, las reuniones y las tareas que no son responsabilidad íntegra del estudiante.

T1.1 Contexto del Trabajo – Entorno de negocio

T1.2 Contexto del Trabajo – Contexto científico-técnico

T1.3 Contexto del Trabajo – Estado del arte

T1.4 Introducción – Planteamiento del problema (*Problem Statement*)

T1.5 Introducción – Objetivos del trabajo

T1.6 Planificación – Metodología de trabajo

T1.7 Planificación – Planificación temporal

T1.8 Planificación – Presupuesto

T1.9 Finalización de la Memoria *Sprint* #1

T1.10 Finalización de la Presentación *Sprint* #1

T1.11 Entrega Final *Sprint* #1

En la Tabla 2.2 se puede observar la relación entre las tareas y los subobjetivos de este *Sprint* expresados en forma de matriz. Se considera preciso explicar que se obvian los subobjetivos "Planificación del *Sprint*" y "Desarrollo del *Sprint*" debido a que las tareas asociadas a ellos también se han obviado en la lista de tareas.

Subobjetivos / Tareas	S1.1	S1.2	S1.3	S1.4	S1.5	S1.6	S1.9
T1.1		X			X	X	
T1.2			X		X	X	
T1.3				X	X	X	
T1.4	X				X	X	
T1.5	X				X	X	
T1.6	X				X	X	
T1.7	X				X	X	
T1.8	X				X	X	
T1.9					X		
T1.10						X	
T1.11							X

Tabla 2.2: Matriz de relación entre tareas y subobjetivos del *Sprint* #1

En la Figura 2.2 se puede observar el diagrama de *Gantt* de este *Sprint*.

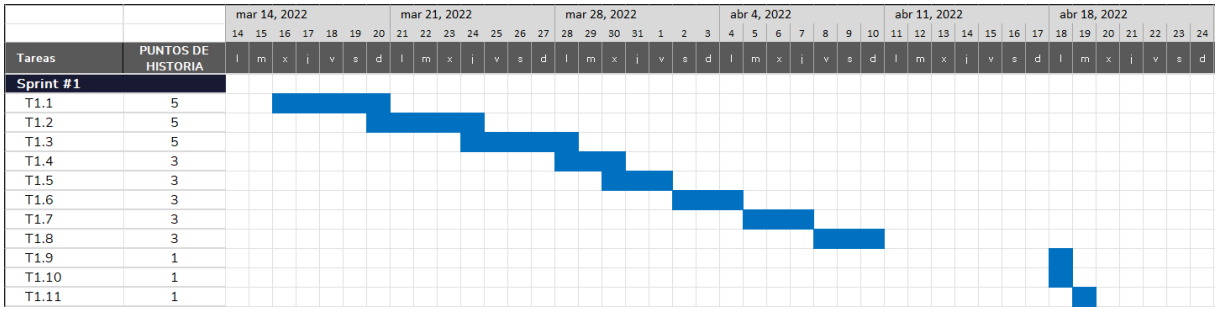


Figura 2.2: Planificación temporal *Sprint* #1

2.2.2. *Sprint* #2

El *Sprint* #2 lo define un objetivo del *Sprint* que comprende 11 subobjetivos, que se abordarán mediante la realización de 13 tareas.

Subobjetivos Los subobjetivos de este *Sprint* son:

S2.1 Caracterización del Proyecto

S2.2 Entorno de negocio

S2.3 Contexto científico-técnico

S2.4 Estado del arte

S2.5 Análisis. Comprensión y definición de lo que tiene que hacer el sistema informático que se va a desarrollar en este proyecto, así como las condiciones en las que debe prestar esta funcionalidad.

S2.6 Diseño. Descripción precisa del *software* (que sirva como base para su implementación), a partir de los requisitos especificados durante el análisis.

S2.7 Memoria *Sprint* #2

S2.8 Presentación *Sprint* #2

S2.9 Planificación del *Sprint*

S2.10 Desarrollo del *Sprint*

S2.11 Finalización del *Sprint*

Tareas A continuación, se listan las tareas de este *Sprint*, a excepción de las tareas colectivas, esto es, las reuniones y las tareas que no son responsabilidad íntegra del estudiante.

T2.1 Introducción (Corrección)

T2.2 Contexto del Trabajo (Corrección)

T2.3 Planificación – Metodología de trabajo

T2.4 Planificación – Planificación temporal

T2.5 Planificación – Presupuesto

T2.6 Análisis – Requisitos de usuario

T2.7 Análisis – Requisitos funcionales

T2.8 Análisis – Atributos de calidad y restricciones

T2.9 Diseño – Arquitecturas

T2.10 Diseño – Diagramas e interfaces de usuario

T2.11 Finalización de la Memoria *Sprint* #2

T2.12 Finalización de la Presentación *Sprint* #2

T2.13 Entrega Final *Sprint* #2

En la Tabla 2.3 se puede observar la relación entre las tareas y los subobjetivos de este *Sprint* expresados en forma de matriz. Se considera preciso explicar que se obvian los subobjetivos "Planificación del *Sprint*" y "Desarrollo del *Sprint*" debido a que las tareas asociadas a ellos también se han obviado en la lista de tareas.

Subobjetivos / Tareas	S2.1	S2.2	S2.3	S2.4	S2.5	S2.6	S2.7	S2.8	S2.11
T2.1	X						X	X	
T2.2		X	X	X			X	X	
T2.3	X						X	X	
T2.4	X						X	X	
T2.5	X						X	X	
T2.6					X		X	X	
T2.7					X		X	X	
T2.8					X		X	X	
T2.9						X	X	X	
T2.10						X	X	X	
T2.11							X		
T2.12								X	
T2.13									X

Tabla 2.3: Matriz de relación entre tareas y subobjetivos del *Sprint* #2

En la Figura 2.3 se puede observar el diagrama de *Gantt* de este *Sprint*.

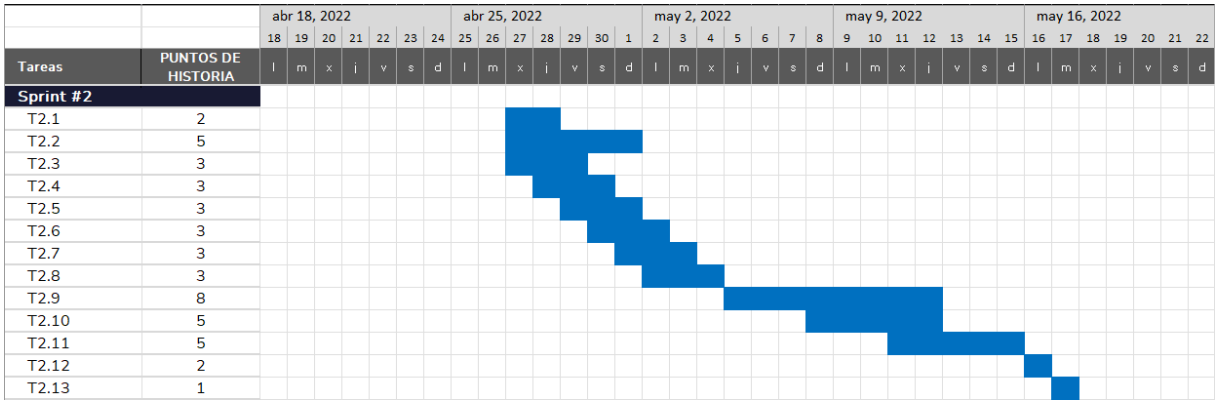


Figura 2.3: Planificación temporal *Sprint* #2

2.2.3. *Sprint* #3

El *Sprint* #3 lo define un objetivo del *Sprint* que comprende 11 subobjetivos, que se abordarán mediante la realización de 15 tareas.

Subobjetivos Los subobjetivos de este *Sprint* son:

S3.1 Caracterización del Proyecto

S3.2 Entorno de negocio

S3.3 Contexto científico-técnico

S3.4 Estado del arte

S3.5 Análisis

S3.6 Diseño

S3.7 Memoria *Sprint* #3

S3.8 Presentación *Sprint* #3

S3.9 Planificación del *Sprint*

S3.10 Desarrollo del *Sprint*

S3.11 Finalización del *Sprint*

Tareas A continuación, se listan las tareas de este *Sprint*, a excepción de las tareas colectivas, esto es, las reuniones y las tareas que no son responsabilidad íntegra del estudiante.

T3.1 Introducción - Planteamiento del problema (Corrección)

T3.2 Planificación - Presupuesto (Corrección)

T3.3 Contexto del Trabajo - Entorno de negocio (Corrección)

T3.4 Contexto del Trabajo - Contexto científico-técnico (Corrección)

T3.5 Contexto del Trabajo - Estado del arte (Corrección)

T3.6 Planificación - Metodología de trabajo (Corrección)

T3.7 Planificación - Planificación temporal (Corrección)

T3.8 Análisis - Requisitos de usuario (Corrección)

T3.9 Análisis - Requisitos funcionales (Corrección)

T3.10 Análisis - Atributos de calidad y restricciones (Corrección)

T3.11 Diseño - Arquitecturas (Corrección)

T3.12 Diseño - Diagramas e interfaces de usuario

T3.13 Finalización de la Memoria *Sprint* #3

T3.14 Finalización de la Presentación *Sprint* #3

T3.15 Entrega Final *Sprint* #3

En la Tabla 2.4 se puede observar la relación entre las tareas y los subobjetivos de este *Sprint* expresados en forma de matriz. Se considera preciso explicar que se obvian los subobjetivos "Planificación del *Sprint*" y "Desarrollo del *Sprint*" debido a que las tareas asociadas a ellos también se han obviado en la lista de tareas.

Subobjetivos / Tareas	S3.1	S3.2	S3.3	S3.4	S3.5	S3.6	S3.7	S3.8	S3.11
T3.1	X						X	X	
T3.2	X						X	X	
T3.3		X					X	X	
T3.4			X				X	X	
T3.5				X			X	X	
T3.6	X						X	X	
T3.7	X						X	X	
T3.8					X		X	X	
T3.9					X		X	X	
T3.10					X		X	X	
T3.11						X	X	X	
T3.12						X	X	X	
T3.13							X		
T3.14								X	
T3.15									X

Tabla 2.4: Matriz de relación entre tareas y subobjetivos del *Sprint* #3

En la Figura 2.4 se puede observar el diagrama de *Gantt* de este *Sprint*.

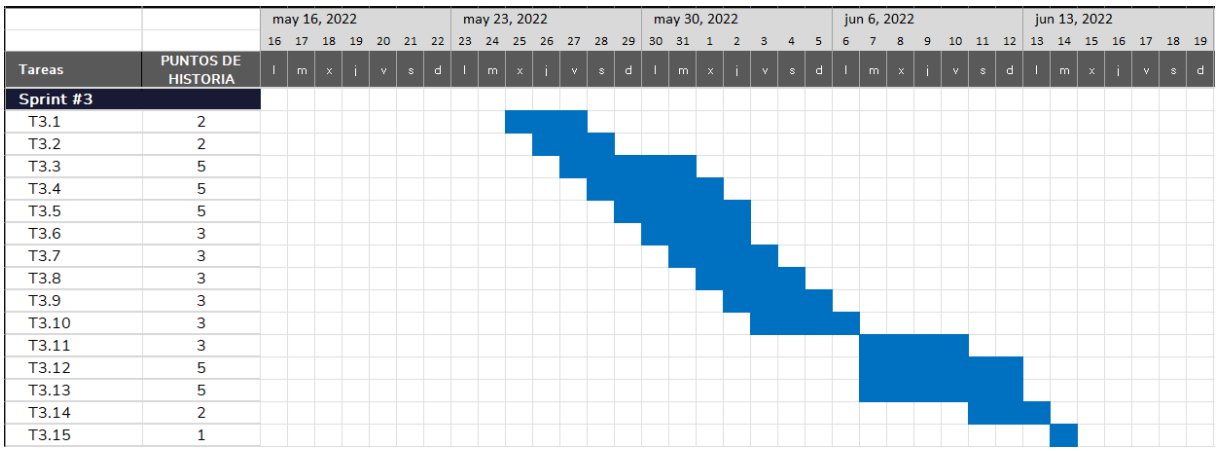


Figura 2.4: Planificación temporal *Sprint* #3

2.2.4. *Sprint* #4

El *Sprint* #4 lo define un objetivo del *Sprint* que comprende 14 subobjetivos, que se abordarán mediante la realización de 11 tareas.

Subobjetivos Los subobjetivos de este *Sprint* son:

S4.1 Caracterización del Proyecto

S4.2 Entorno de negocio

S4.3 Contexto científico-técnico

S4.4 Estado del arte

S4.5 Análisis

S4.6 Diseño

S4.7 Implementación. Construcción del sistema informático objeto del proyecto, es decir, implementación de los casos de uso requeridos para proporcionar la funcionalidad deseada.

S4.8 Prueba. Evaluación de que el sistema informático construido se ajusta a los requisitos del proyecto. Concretamente, evaluar que los datos sintéticos generados se ajustan a lo esperado y que el *dashboard* ofrece la funcionalidad necesaria para poder analizar (visualmente) su calidad.

S4.9 Tuning. Mejora de la calidad de los datos sintéticos generados, evaluando diferentes valores para los diferentes parámetros considerados en el proceso de generación de los datos.

S4.10 Memoria *Sprint* #4

S4.11 Presentación *Sprint* #4

S4.12 Planificación del *Sprint*

S4.13 Desarrollo del *Sprint*

S4.14 Finalización del *Sprint*

Tareas A continuación, se listan las tareas de este *Sprint*, a excepción de las tareas colectivas, esto es, las reuniones y las tareas que no son responsabilidad íntegra del estudiante.

T4.1 Introducción (Corrección)

T4.2 Planificación (Corrección)

T4.3 Contexto del Trabajo (Corrección)

T4.4 Análisis (Corrección)

T4.5 Diseño - Diagramas e interfaces de usuario

T4.6 Implementación CU-01

T4.7 Prueba

T4.8 *Tuning*

T4.9 Finalización de la Memoria *Sprint* #4

T4.10 Finalización de la Presentación *Sprint* #4

T4.11 Entrega Final *Sprint* #4

En la Tabla 2.5 se puede observar la relación entre las tareas y los subobjetivos de este *Sprint* expresados en forma de matriz. Se considera preciso explicar que se obvian los subobjetivos "Planificación del *Sprint*" y "Desarrollo del *Sprint*" debido a que las tareas asociadas a ellos también se han obviado en la lista de tareas.

S. / T.	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	4.10	4.11	4.14
4.1	X									X	X	
4.2	X									X	X	
4.3		X	X	X						X	X	
4.4					X					X	X	
4.5						X				X	X	
4.6							X			X	X	
4.7								X		X	X	
4.8									X	X	X	
4.9										X		
4.10											X	
4.11												X

Tabla 2.5: Matriz de relación entre tareas y subobjetivos del *Sprint* #4

En la Figura 2.5 se puede observar el diagrama de *Gantt* de este *Sprint*.

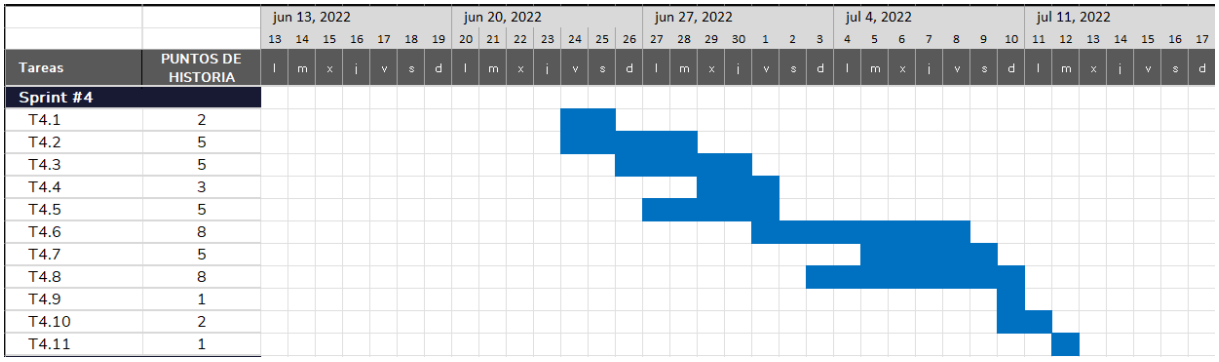


Figura 2.5: Planificación temporal *Sprint* #4

2.2.5. *Sprint* #5

El *Sprint* #5 lo define un objetivo del *Sprint* que comprende subobjetivos, que se abordarán mediante la realización de tareas.

Subobjetivos Los subobjetivos de este *Sprint* son:

S5.1 Caracterización del Proyecto

S5.2 Entorno de negocio

S5.3 Contexto científico-técnico

S5.4 Estado del arte

S5.5 Análisis

S5.6 Diseño

S5.7 Implementación

S5.8 Prueba

S5.9 *Tuning*

S5.10 Diseño Experimental. Elección de las métricas que evalúan el desempeño del generador de datos sintéticos es decir, dichas métricas evalúan la similitud de los datos generados respecto a un conjunto de datos originales. Este subobjetivo incluye también la búsqueda de un conjunto de datos real para realizar la comparación anterior.

S5.11 Experimentación y Análisis de Resultados. Realización de pruebas de ejemplo del sistema y muestra de los resultados obtenidos.

S5.12 Discusión de Resultados. Explicación de los resultados obtenidos en las pruebas de ejemplo del sistema.

S5.13 Memoria *Sprint* #5

S5.14 Presentación *Sprint* #5

S5.15 Planificación del *Sprint*

S5.16 Desarrollo del *Sprint*

S5.17 Finalización del *Sprint*

Tareas A continuación, se listan las tareas de este *Sprint*, a excepción de las tareas colectivas, esto es, las reuniones y las tareas que no son responsabilidad íntegra del estudiante.

T5.1 Análisis (Corrección)

T5.2 Diseño (Corrección)

T5.3 Diseño - Diagramas e interfaces de usuario

T5.4 Implementación CU-01 (Corrección)

T5.5 Implementación CU-02

T5.6 Implementación CU-03

T5.7 Implementación - Interfaz de usuario

T5.8 Prueba

T5.9 *Tuning*

T5.10 Experimentación y evaluación - Diseño Experimental

T5.11 Experimentación y evaluación - Experimentación y Análisis de Resultados

T5.12 Experimentación y evaluación - Discusión de Resultados

T5.13 Introducción (Corrección)

T5.14 Planificación (Corrección)

T5.15 Contexto del Trabajo (Corrección)

T5.16 Conclusiones

T5.17 Trabajo futuro

T5.18 Finalización de la Memoria *Sprint* #5

T5.19 Finalización de la Presentación *Sprint* #5

T5.20 Entrega Final *Sprint* #5

En la Tabla 2.6 se puede observar la relación entre las tareas y los subobjetivos de este *Sprint* expresados en forma de matriz. Se considera preciso explicar que se obvian los subobjetivos "Planificación del *Sprint*" y "Desarrollo del *Sprint*" debido a que las tareas asociadas a ellos también se han obviado en la lista de tareas.

S. / T.	5.1	5.2	5.3	5.4	5.5	5.6	5.7	5.8	5.9	5.10	5.11	5.12	5.13	5.14	5.17
5.1					X								X	X	
5.2						X							X	X	
5.3						X							X	X	
5.4							X								
5.5							X								
5.6							X								
5.7							X								
5.8								X							
5.9									X						
5.10										X			X	X	
5.11											X		X	X	
5.12												X	X	X	
5.13	X												X	X	
5.14	X												X	X	
5.15		X	X	X									X	X	
5.16													X	X	
5.17													X	X	
5.18													X		
5.19														X	
5.20															X

Tabla 2.6: Matriz de relación entre tareas y subobjetivos del *Sprint* #5

En la Figura 2.6 se puede observar el diagrama de *Gantt* de este *Sprint*.

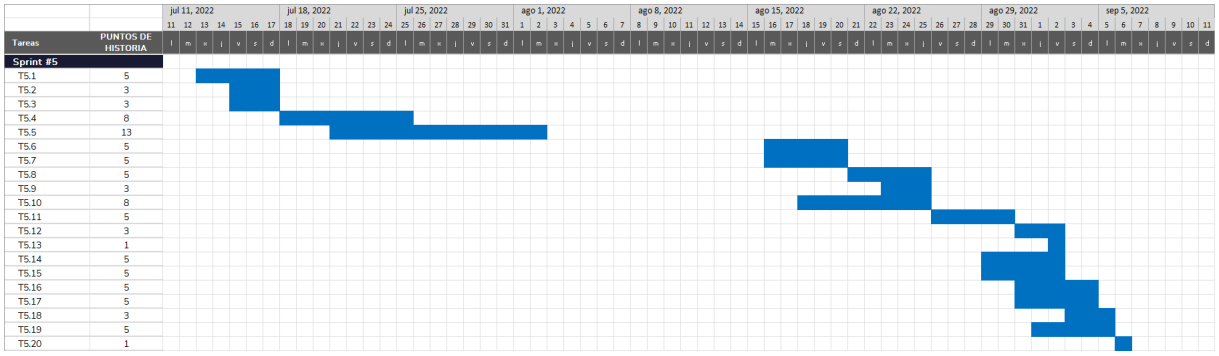


Figura 2.6: Planificación temporal *Sprint* #5

2.3. Presupuesto

Tras haber realizado la planificación temporal, es necesario estimar los costes del proyecto. Realizar un presupuesto inicial del proyecto que se corresponda realmente con los costes reales al final de éste resulta una tarea especialmente complicada al tratarse de un proyecto de investigación. Esto es debido, principalmente, a que no se saben exactamente las herramientas que se van a necesitar, o los contratiempos que pueden afectar directamente a las estimaciones iniciales (nuevas ideas surgidas tras el descubrimiento de algún dato, bloqueos por la inexistencia de documentación o por la ausencia de experiencia previa con la tarea, etc.).

A continuación, se presenta el presupuesto desglosado en tres categorías (*hardware*, *software* y Recursos Humanos (RRHH)); se corresponden con la estimación inicial realizada al comienzo del proyecto. Cabe destacar que, para calcular el coste real asociado al uso del *hardware* y del *software*, se deben usar unas fórmulas que tienen en cuenta la frecuencia de uso, el tiempo utilizado, y el coste mensual de estas herramientas. Éstas se formulan a continuación junto con las unidades de cada magnitud entre paréntesis:

$$\text{Coste por mes}\left(\frac{€}{\text{mes}}\right) = \frac{\text{Coste total}\ (€)}{\text{Vida útil (meses)}} \tag{2.1}$$

$$\text{Coste real por mes}\left(\frac{€}{\text{mes}}\right) = \text{Coste por mes}\left(\frac{€}{\text{mes}}\right) \times \text{Porcentaje de uso} \tag{2.2}$$

$$\text{Coste real}\ (€) = \text{Coste real por mes}\left(\frac{€}{\text{mes}}\right) \times \text{Tiempo de uso (meses)} \tag{2.3}$$

2.3.1. *Hardware*

El desarrollo del proyecto se llevará a cabo en un ordenador personal de gama media-alta cuyas especificaciones son las siguientes: procesador i7-1065G7, sistema operativo de 64 bits, 16 GB de RAM, 256 GB de memoria SSD, y 1 TB de memoria HDD. Además, para conseguir una correcta comunicación con el resto del equipo UVAGILE, poder consultar documentación, emplear herramientas *online*, etc., es necesario disponer de una buena conexión a Internet. Por ello, este gasto también se incluye en el presupuesto. Cabe mencionar que el porcentaje de uso tan bajo de este recurso se debe a que se comparte *Wi-Fi* con otros dos estudiantes, por lo que se parte de un 33,3 % de uso total al mes por parte del trabajador.

El resto de los gastos corrientes, como la electricidad para cargar el ordenador, no se consideran para el cálculo debido al desconocimiento de su porcentaje de uso, ya que no se puede discernir del consumo no atribuible al proyecto y, por tanto, no es medible en el ámbito de éste.

Por todo ello, el coste total asociado al *hardware* es de 79,45 € y se desglosa según las especificaciones de la Tabla 2.7.

Herramienta	Coste total (€)	Vida útil (meses)	Uso (%)	Uso (meses)	Coste (€)
Ordenador personal	899	$5 \cdot 12 = 60$	80	4	47,95
Conexión a Internet	31,5 (al mes)	-	25	4	31,5
				Total:	79,45

Tabla 2.7: Presupuesto *hardware*

2.3.2. *Software*

El coste total asociado al *software* es de 0 € ya que todas las herramientas utilizadas tienen licencia gratuita para estudiantes como se puede observar en la Tabla 2.8.

Herramienta	Coste por mes (€)	Uso (%)	Uso (meses)	Coste real (€)
Microsoft Teams	0	80	4	0
Trello	0	100	4	0
EasyRetro	0	100	4	0
Overleaf	0	50	4	0
Mendeley	0	40	4	0
Jupyter Notebook	0	90	4	0
			Total:	0

Tabla 2.8: Presupuesto *software*

2.3.3. Recursos humanos

A pesar de que el trabajo va a ser realizado por una única persona, ésta tomará diferentes roles a lo largo del proyecto (analista de sistema, arquitecto de *software*, desarrollador Python y *tester software*).

Por esta razón, su sueldo bruto será la suma del sueldo bruto obtenido por cada rol adoptado [3], es decir, 5684,1 €, como indica el total de la Tabla 2.9.

Rol	Sueldo (€/hora)	Tiempo (horas)	Total (€)
Analista de sistema (30 %)	18,3	90	1647
Arquitecto de *software* (20 %)	26,71	60	1602,6
Desarrollador Python (40 %)	16,64	120	1996,8
Tester software (10 %)	14,59	30	437,7
		Total:	5684,1

Tabla 2.9: Sueldos (planificación)

Además, dado que es necesario dar de alta en la Seguridad Social a esta persona, se debe tener en cuenta este coste adicional, que será de $0{,}309 \cdot (1647 + 1602{,}6 + 1996{,}8 + 437{,}7)$ € = $0{,}309 \cdot 5684{,}1$ € = 1756,4 €, ya que se corresponde con el 30,9 % del sueldo bruto de la persona, al tratarse de un contrato indefinido [26]. Por todo ello, el coste total asociado al personal es de 7440,5 € y se desglosa según lo explicado en la Ecuación 2.4.

$$\text{Presupuesto RRHH} = 5684{,}1\,€\ \text{(Sueldo)} + 1756{,}4\,€\ \text{(Cotización)} = 7440{,}5\,€ \tag{2.4}$$

[3] Todos los sueldos de la Tabla 2.9 se han hallado teniendo en cuenta los datos proporcionados por el buscador de salario bruto Glassdoor [24], todos excepto el de analista de sistema, para el cual se ha utilizado el buscador de LinkedIn [25]. En todos los casos se ha considerado un trabajador en Madrid, ya que es un lugar muy genérico y que recoge unos salarios estándar a nivel europeo. También se ha tenido en cuenta que un trabajador medio realiza un total de aproximadamente 1800 horas al año.

A partir de este estudio se prevé que el coste total del proyecto es de 7519,95 €, de los cuales 79,45 € derivan del uso de recursos *hardware*, y 7440,5 € de costes de personal, como se indica en la Tabla 2.10.

Hardware (€)	*Software* (€)	RRHH (€)	Total (€)
79,45	0	7440,5	7519,95

Tabla 2.10: Presupuesto

2.4. Balance temporal y económico

En esta sección, se lleva a cabo un balance del proyecto, esto es, un análisis de la diferencia temporal y económica entre la realidad y la planificación. Puesto que se trata de una comparación con lo planificado, el balance temporal se estudia tras el final de cada *Sprint* mientras que el económico se hace de manera global al final del proyecto.

2.4.1. Balance temporal

En el caso de este proyecto, finalmente se realizan 5 *Sprints* con una duración de 4 semanas cada uno con la siguiente cronología:

- Primer *Sprint* (16 mar - 19 abr, con descanso en el periodo 6 abr - 19 abr)
- Segundo *Sprint* (20 abr - 17 may)
- Tercer *Sprint* (18 may - 14 jun)
- Cuarto *Sprint* (15 jun - 12 jul)
- Quinto *Sprint* (13 jul - 6 sep, con descanso en el periodo 3 ago - 30 ago)

A pesar de que todos los *Sprints* han de tener la misma duración, el segundo *Sprint* tiene un desfase de dos semanas debido a las vacaciones de Semana Santa. Lo mismo sucede con el quinto *Sprint*, con un desfase de tres semanas debido a las vacaciones de verano que suceden en el mes de agosto.

En esta subsección, se va a mostrar el desempeño temporal real que el estudiante/empleado responsable de la consecución del producto objetivo de este proyecto ha dedicado a cada una de las tareas planificadas para cada *Sprint*, mostrando así los posibles retrasos y atascos que han dificultado el avance y la secuencia normal de cada *Sprint*. Para ello, se muestra un diagrama de *Gantt* en el que se indica, para cada tarea, un esquema de colores que representan lo siguiente: en azul, los días planificados para realizar la tarea; en verde, los días dedicados a la tarea que coinciden con lo que se había planificado; y en rojo, los días dedicados a la tarea que no coinciden con lo que se había planificado, ya sea antes o después.

Se considera oportuno comentar que el estudiante cursó, al mismo tiempo que este TFG, tres asignaturas del grado y otro TFG, por lo que hubo temporadas en las que no pudo dedicar el tiempo planeado (75 horas por *Sprint* en 4 *Sprints*) y, como consecuencia, quedaron tareas planificadas sin realizar o realizadas parcialmente. Esto explica también la necesidad de añadir el quinto *Sprint*.

Sprint #1

En la Figura 2.7 se puede observar el balance temporal de este *Sprint*. Las tareas T1.5 **T1.5**, T1.7 **T1.7** y T1.8 **T1.8** no se han podido llevar a cabo.

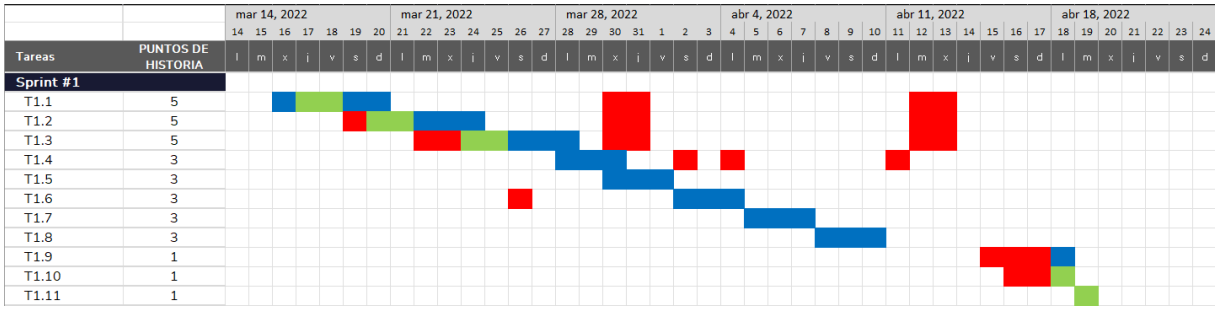


Figura 2.7: Balance temporal *Sprint* #1

Sprint #2

En la Figura 2.8 se puede observar el balance temporal de este *Sprint*. La tarea T2.10 **T2.10** no se ha podido llevar a cabo.

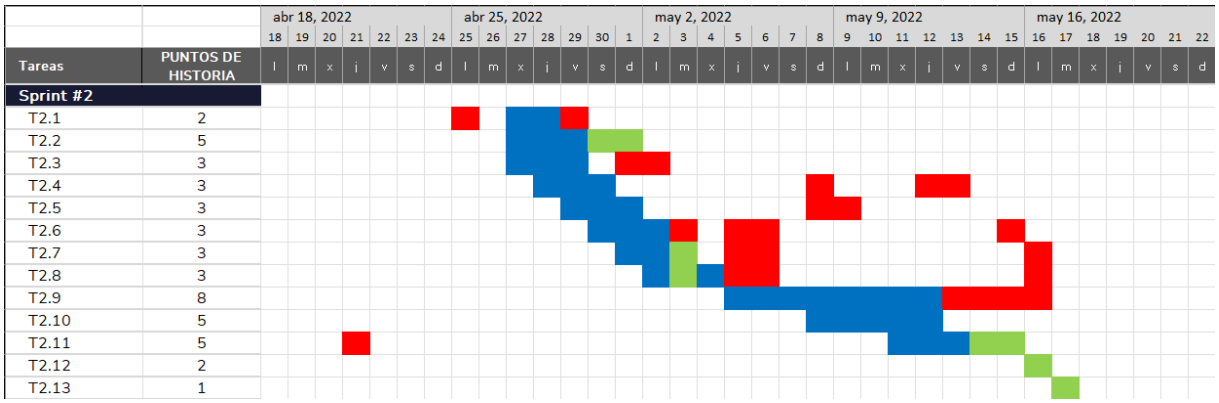


Figura 2.8: Balance temporal *Sprint* #2

Sprint #3

En la Figura 2.9 se puede observar el balance temporal de este *Sprint*. La tarea T3.6 **T3.6** no se ha podido llevar a cabo.

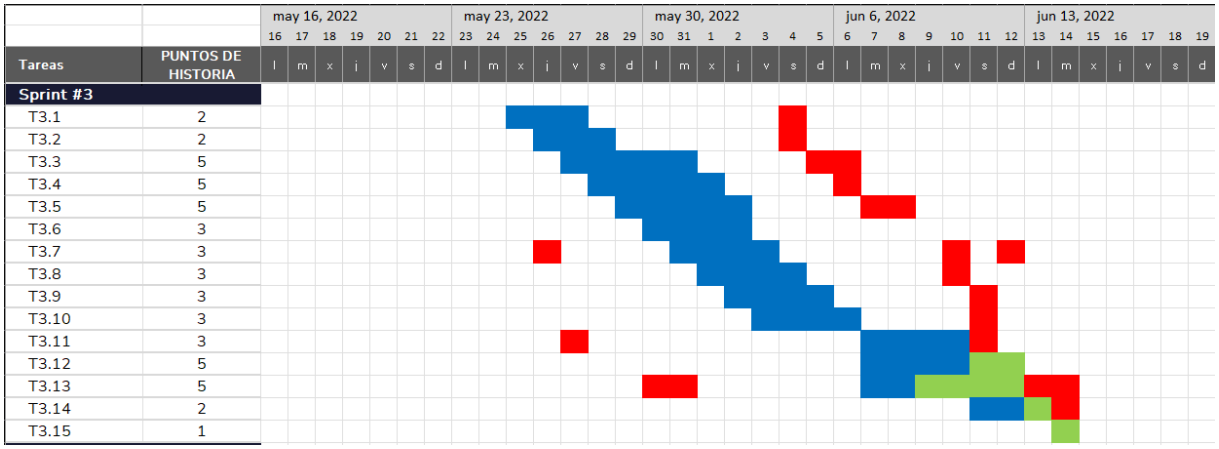


Figura 2.9: Balance temporal *Sprint* #3

Sprint #4

En la Figura 2.10 se puede observar el balance temporal de este *Sprint*. Las tareas T4.5 **T4.5**, T4.7 **T4.7** y T4.8 **T4.8** no se ha podido llevar a cabo.

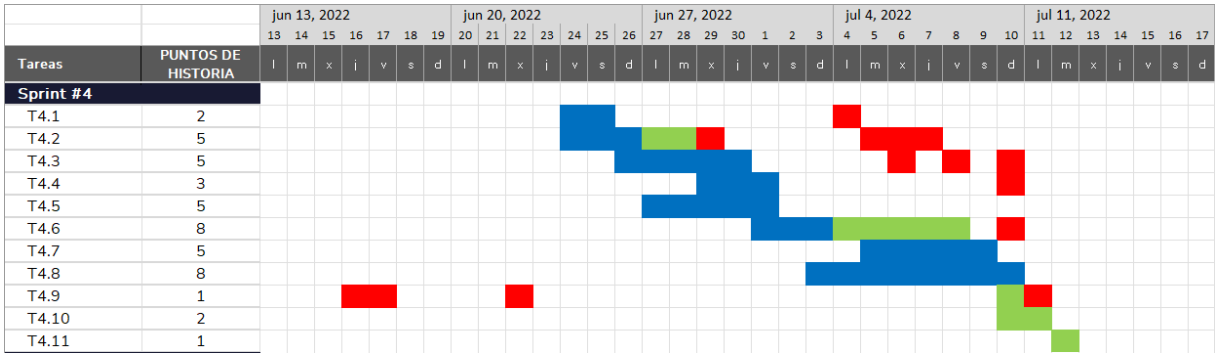


Figura 2.10: Balance temporal *Sprint* #4

Sprint #5

En la Figura 2.11 se puede observar el balance temporal de este *Sprint*. Se han podido llevar a cabo todas las tareas, lo que completa el producto objetivo del proyecto.

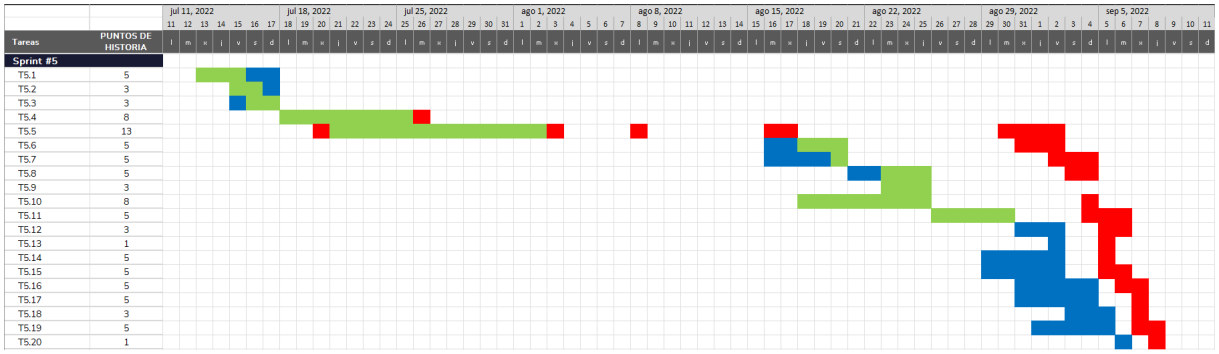


Figura 2.11: Balance temporal *Sprint* #5

La densidad de trabajo que se sufre durante los últimos meses del curso académico (abril, mayo y junio), hizo que hubiera retrasos en las primeras tareas del proyecto. Estas tareas iniciales consistían en buscar información sin la cual no se podía ni plantear el sistema *software* protagonista de este trabajo. Por tanto, esto retrasó las tareas de implementación, objetivo que no pudo plantearse hasta el cuarto *Sprint*. Dado que, incluso las dos primeras semanas de este cuarto *Sprint* seguía habiendo actividad académica relacionada con el resto de asignaturas matriculadas, se consideró inviable la idea de incluir todos los objetivos restantes del proyecto en el objetivo de ese "último" *Sprint*. Esto desencadenó la planificación de un quinto *Sprint* en el que se pudieron finalizar todos los objetivos del proyecto. Además, como puede observarse al comparar la Figura 2.11 con el resto de diagramas de *Gantt* de esta sección, el cumplimiento temporal de las tareas mejora. Esta mejora es debida a que se gana experiencia en la tarea de planificación de tareas tras repetirla al principio de cada *Sprint* y a una mayor disponibilidad temporal para la realización del TFG. Para finalizar, cabe comentar que los únicos retrasos acaecidos en la planificación del último *Sprint* se producen tras la *reunión de sincronización* celebrada el 30 de agosto, en la que se encuentran una serie de errores en las tareas ya realizadas que se deben de corregir con cierta prioridad.

2.4.2. Balance económico

Los costes iniciales de *software* no han sufrido modificaciones debido a que se utilizan herramientas de *software* libre. Sin embargo, el número de horas añadidas debido a la necesidad de un quinto *Sprint* sí ha modificado el presupuesto *hardware* y los costes de personal, ya que ha supuesto otro mes de trabajo (no se tienen en cuenta los tres meses de vacaciones de verano de agosto).

La tabla de presupuesto *hardware* (Tabla 2.7) se ve modificada como se observa en la Tabla 2.11. Por tanto, el coste total real asociado al *hardware* es de 99,31 € y se desglosa según las especificaciones de la Tabla 2.11. Así que la diferencia entre la planificación y la realidad es de $99{,}31\,€ - 79{,}45\,€ = 19{,}86\,€$.

Herramienta	Coste total (€)	Vida útil (meses)	Uso (%)	Uso (meses)	Coste (€)
Ordenador personal	899	$5 \cdot 12 = 60$	80	5	59,93
Conexión a Internet	31,5 (al mes)	-	25	5	39,38
				Total:	99,31

Tabla 2.11: Balance económico *hardware*

Para cuantificar de una forma más estricta la diferencia de coste que se ha producido en cuanto al presupuesto de RRHH, se va a tener en cuenta el número exacto de horas que ha trabajado el estudiante, contabilizado mediante el artefacto "Cuaderno de trabajo" 2.1.3. En total, el estudiante ha trabajado 346 horas. La tabla de presupuesto RRHH (Tabla 2.9) se ve modificada como se observa en la Tabla 2.12.

Por esta razón, su sueldo bruto será la suma del sueldo bruto obtenido por cada rol adoptado, es decir, 5684,1 €, como indica el total de la Tabla 2.12.

Rol	Sueldo (€/hora)	Tiempo (horas)	Total (€)
Analista de sistema (30 %)	18,3	103,8	1899,54
Arquitecto de *software* (20 %)	26,71	69,2	1848,33
Desarrollador Python (40 %)	16,64	138,4	2302,98
Tester software (10 %)	14,59	34,6	504,81
		Total:	6555,66

Tabla 2.12: Sueldos (balance)

Además, dado que es necesario dar de alta en la Seguridad Social a esta persona, se debe tener en cuenta este coste adicional, que será de $0{,}309 \cdot (1899{,}54 + 1848{,}33 + 2302{,}98 + 504{,}81)$ € = $0{,}309 \cdot 6555{,}66$ € = 2025,7 €. Por todo ello, el coste total real asociado al personal es de 8581,36 € y se desglosa según lo explicado en la Ecuación 2.5.

$$\text{Coste real (RRHH)} = 6555{,}66\,€\ \text{(Sueldo)} + 2025{,}7\,€\ \text{(Cotización)} = 8581{,}36\,€ \tag{2.5}$$

Así que la diferencia entre la planificación y la realidad es de 8581,36 € − 7440,5 € = 1140,86 €.

A partir de este estudio se prevé que el coste total real del proyecto es de 8680,67 €, de los cuales 99,31 € derivan del uso de recursos *hardware*, y 8581,36 € de costes de personal, como se indica en la Tabla 2.13.

Hardware (€)	*Software* (€)	RRHH (€)	Total (€)
99,31	0	8581,36	8680,67

Tabla 2.13: Balance económico

Así que, globalmente, la diferencia entre la planificación y la realidad es de 8680,67 € − 7519, 95 € = 1160,72 €.

Capítulo 3

Generación de datos sintéticos

El proyecto planteado responde a una necesidad existente en la sociedad. Para identificar los requisitos de una propuesta que persiga satisfacerla, es imprescindible caracterizar debidamente los agentes y factores que tienen cierta influencia sobre ella, es decir, situar dicha necesidad en la realidad actual. Esto incluye agentes de interés (*stakeholders*), recursos y herramientas disponibles, o cualquier otra regla de negocio que establezca una ventaja, restricción u orientación de cara a realizar y materializar la propuesta.

En este capítulo, se hace una introducción a los entornos de negocio *data-driven*, se comenta el papel que juegan en la sociedad y los retos a los que se enfrentan (sección 3.1). Se continúa profundizando en el concepto clave de este proyecto, los datos sintéticos, así como en las diferentes técnicas ya existentes para su generación (sección 3.2). Posteriormente, se explora la investigación existente en el área concreta en que se enmarca el proyecto, haciendo una comparación entre las técnicas y tipos de datos sintéticos existentes (sección 3.3) y se descubren las herramientas y recursos existentes en el panorama científico-técnico ya consolidado (sección 3.4).

3.1. Entornos de negocio *data-driven*

Un entorno de negocio *data-driven* es una organización en la que imperan los datos; está imbuida de una extraordinaria cultura alrededor de ellos (cultura *data-driven*) y la toma de decisiones está basada en el óptimo análisis de esta información [27]. El objetivo en este tipo de entornos es transformar los datos en conocimiento para extraer el máximo provecho de ellos.

El hecho de que los datos se estén volviendo cada vez más críticos, ha propulsado que hoy en día las organizaciones dispongan de varias herramientas y tecnologías para aprovechar el poder de estos activos. Las instituciones que se basan en los datos han demostrado ser mejores en eficiencia, ahorro de costes, centradas en el cliente e innovadoras [1].

Para hacer un buen modelo basado en datos es esencial que las organizaciones descubran la naturaleza exacta de los datos que pueden ser valiosos para ellos. Por tanto, deben reconocer sus objetivos y elaborar estrategias en consecuencia [1]. En la visión general del Índice Global de Protección de Datos de 2020 de Dell Technologies [28], se reveló que las organizaciones administraban casi un 40 % más de datos que el año anterior. En los resultados también se mostró un cambio positivo, un incremento en la cantidad de organizaciones (un 80 % en 2019, en comparación con el 74 % en 2018) que consideraban que sus datos eran valiosos y que estaban extrayendo valor o planeaban hacerlo en el futuro. Según el estudio, las organizaciones en 2020

administraban 13, 53 *petabytes* (PB) de datos, un aumento en comparación con el promedio de 9, 70 PB en 2018 y un aumento del 831 % desde que las organizaciones administraban 1, 45 PB en 2016 [29]. Las organizaciones también estaban comenzando a aprovechar estos datos a través del ML y el Aprendizaje Profundo o *Deep Learning* (DL), con el 34 % de los líderes empresariales en Singapur monetizando datos. Las instituciones pueden pasar de simplemente analizar datos a hacer un uso adecuado de ellos con herramientas de ML e IA [30].

El problema de la escasez de datos es muy importante ya que los datos son el núcleo de cualquier proyecto de IA. A menudo, el tamaño de un conjunto de datos influye en los malos resultados en los proyectos y, en general, es una de las razones por las que no se pueden realizar grandes proyectos de IA. Esta escasez es causada por la falta de datos relevantes o porque el proceso de recopilación es demasiado difícil y lento, entre otros motivos. Otro problema que se podría mencionar es que los analistas de proyectos tienden a subestimar la cantidad de datos necesarios para abordar problemas comunes. Asimismo, los modelos de ML supervisado, los cuales se están utilizando con éxito para responder a una amplia gama de desafíos empresariales, requieren muchos datos, y su rendimiento depende en gran medida del tamaño de los datos de entrenamiento disponibles [31].

Sin embargo, ya no se trata solo de la cantidad de datos, sino de la calidad y diversidad de estos datos. Sin diversidad de datos, es fácil que el sesgo de estos se convierta en un problema, lo que conduce a la entrega de productos por debajo de la media y posibles repercusiones legales, debido a la discriminación contra individuos o grupos [30].

El papel de los datos sintéticos en el ML está aumentando rápidamente debido a que los algoritmos de ML se entrenan con cantidades masivas de datos, que podrían ser difíciles de obtener. También puede desempeñar un papel importante en la creación de algoritmos para el reconocimiento de imágenes y tareas similares, que se están convirtiendo en la línea de base para la IA.

Hay varios beneficios adicionales al usar datos sintéticos en el desarrollo del ML. Algunos de ellos son la facilidad en la producción de datos, una vez que se ha establecido un modelo/entorno sintético inicial, la precisión en el etiquetado (que sería costosa o incluso imposible de obtener a mano), la flexibilidad del entorno sintético que se ajustará según sea necesario para mejorar el modelo, y la usabilidad como sustituto de los datos que contienen información sensible [32].

En resumen, el ML es una herramienta que aporta valor en este tipo de entornos basados en los datos. Sin embargo, estos métodos de ML necesitan muchos datos que, por unas u otras circunstancias, podrían no estar a su disposición. Estos datos faltantes, a su vez, pueden ser obtenidos mediante modelos de ML; más concretamente, mediante Red Neuronal o *Neural Network*s (NNs), algoritmos de DL [30]. Como consecuencia, se introducen en la sección 3.2 las técnicas de generación de datos sintéticos como solución a las necesidades de datos en entornos *data-driven*. Pero, primero, se van a comenzar distinguiendo las ramas de la IA (sección 3.1.1) y los distintos modelos de ML utilizados para generar datos sintéticos (subsección 3.1.2).

3.1.1. Ramas de la IA

El día a día de las personas y las organizaciones genera una gran cantidad de datos que, adecuadamente analizados y tratados, permiten tomar mejores decisiones y comprender correctamente el entorno. Esta gran colección de datos se conoce como *Big Data*. Es aquí donde la IA entra en juego, ya que su aplicación ayuda en el trabajo de explotar o aprovechar este gran conjunto de datos [33].

La relación entre la IA y algunas de sus ramas se puede observar en la Figura 3.1. Cada una de estas ciencias se trata, a continuación, dentro de esta subsección.

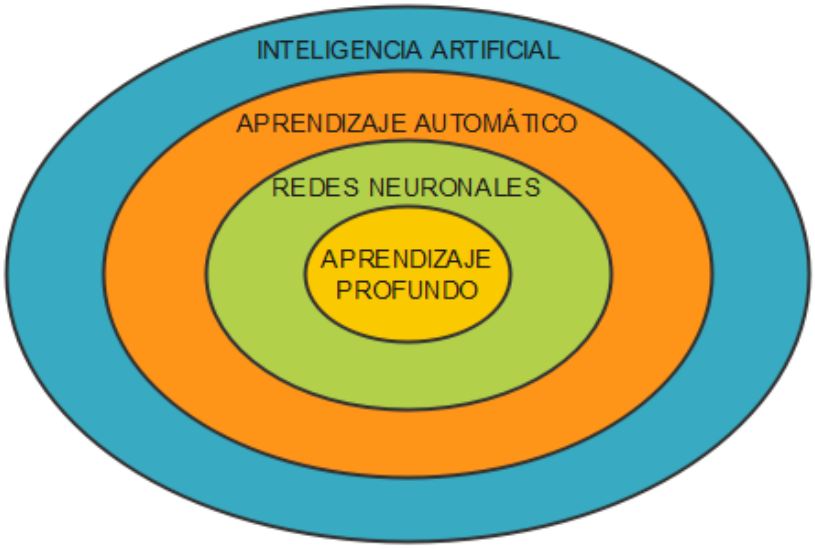


Figura 3.1: Esquema de las ramas de la IA

Las redes neuronales más simples, el perceptrón (solo dos capas, la de entrada y la de salida), se definieron por primera vez en 1943. No fue hasta 1990 que aparecieron las primeras convoluciones (es un tipo de red neuronal) y el reconocimiento de texto. Desde entonces y hasta el año 2012 se produjo una temporada de sequía en este terreno conocida como "Invierno de la IA". A partir de entonces comienza su "Renacimiento" hasta el día de hoy.

Inteligencia artificial

Aunque han surgido varias definiciones en las últimas décadas, John McCarthy define la IA como [34]: "La ciencia y la ingeniería de crear máquinas inteligentes, especialmente programas informáticos inteligentes. Está relacionada con la tarea similar de utilizar ordenadores para comprender la inteligencia humana, pero la IA no se limita a métodos que sean observables biológicamente" [35, 36].

Aprendizaje automático

En esta subsección, se explican los tipos de algoritmos de ML que existen, los usos de esta ciencia y las etapas de su proceso.

Tipos y usos El ML alberga algoritmos que, principalmente, se pueden clasificar en las siguientes categorías [37]:

- Aprendizaje supervisado (*supervised learning*)
  - Se dispone de datos de ejemplos y sus resultados (datos etiquetados).
  - Su objetivo es predecir la respuesta correcta para los nuevos ejemplos desconocidos.

- Aprendizaje no supervisado (*unsupervised learning*)
  - Se dispone de datos de ejemplo, pero no se conoce su resultado (datos sin etiquetar).
  - Su objetivo es encontrar patrones en los datos (agrupamientos "naturales" de los datos).
- Aprendizaje por refuerzo (*reinforcement learning*)
  - Se realizan acciones y se obtienen recompensas.
  - Su objetivo es aprender comportamientos que reporten mayores recompensas.

El ML tiene multitud de usos. Entre las tareas superadas por los modelos de ML cabe destacar la construcción de bases de conocimiento a partir de experiencia y observaciones, la clasificación y diagnóstico, la minería de datos, el descubrimiento de modelos matemáticos que explican un fenómeno a partir de una gran cantidad de datos, etc. [38]

Etapas del proceso Todo proyecto de ML requiere de un minucioso proceso en el que cada etapa supone un avance respecto a la anterior. Las principales fases de implementación de un modelo de ML [39] se pueden observar en el diagrama de flujo de la Figura 3.2 y en la representación de la Figura 3.3.

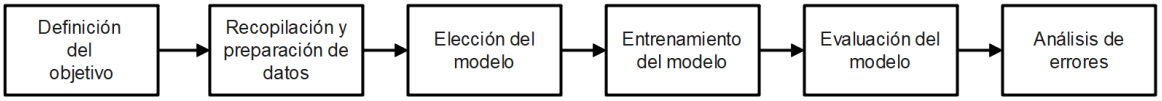


Figura 3.2: Diagrama de flujo de las fases de un proyecto de ML

- Definición del objetivo. Consiste en plantear un problema que requiera una solución a medio-largo plazo. Existen diferentes formas de estructurar el objetivo de un proyecto en el que se va a implementar tecnología ML pero siempre hay que establecerlo en base a las necesidades de negocio y a las posibilidades que tiene la compañía en función de los **datos** de los que dispone.
- Recopilación y preparación de datos. Para que un *software* que utiliza ML funcione correctamente, debe nutrirse de una gran cantidad de **datos** que servirán como punto de partida; después, la máquina continuará su aprendizaje en base a los nuevos datos que vaya extrayendo y procesando. Es más importante la calidad de los datos que la cantidad. No obstante, lo idóneo es plantear un equilibrio ya que cuantos más y mejores datos haya para empezar, mejor será el rendimiento del modelo. La preparación de los datos para un proyecto de ML es un proceso largo y tedioso en el que el analista de datos se enfrenta a grandes retos. Los datos tienen que ser analizados y procesados correctamente para evitar resultados engañosos. En ocasiones, hay irregularidades que deben ser detectadas y corregidas ya que afectan a la calidad de la información. Es habitual encontrar datos incompletos, ruidosos y/o incoherentes. Por este motivo, las tareas de procesamiento y limpieza deben de estar bien efectuadas antes de la puesta en marcha del modelo.

- Elección del modelo. Consiste en seleccionar un modelo/algoritmo de ML que encaje con el objetivo.

- Entrenamiento del modelo. Consiste en suministrar la información que permita que el algoritmo de ML haga su aprendizaje inicial. En esta fase, los **datos** deben estar totalmente contrastados y albergar las respuestas correctas (solo en el caso del ML supervisado), también conocidas como atributos de destino. De esta forma, el algoritmo de aprendizaje es capaz de plantear correlaciones en los datos de entrenamiento y proporcionar así un modelo que almacena dichas correlaciones.

- Evaluación del modelo. Cerciorarse de que el modelo funciona correctamente es importante y permite comprobar la validez del proyecto. Para poder evaluar un modelo correctamente se debe separar el conjunto de **datos** en dos partes bien diferenciadas. Por un lado, la muestra de datos de prueba o *test* y, por otro, la fuente de datos de entrenamiento. Hay que adjuntar los datos al modelo elegido y comparar las predicciones devueltas con el valor real que se espera (objetivo de aprendizaje). Además, se debe diseñar una métrica que indique la efectividad de la predicción y la coincidencia de valores.

- Análisis de errores. Tras la puesta en marcha de un modelo de ML pueden darse múltiples situaciones que den lugar a equívocos que no se habían tenido en cuenta. Por ello, el análisis de errores es la última fase y permite modelar y cambiar los aspectos no relevantes para el proyecto. Este análisis permite mejorar el rendimiento del modelo y profundizar aún más en las fases previas a la implementación y entrenamiento, generando nuevas conclusiones sobre qué necesita el proyecto.

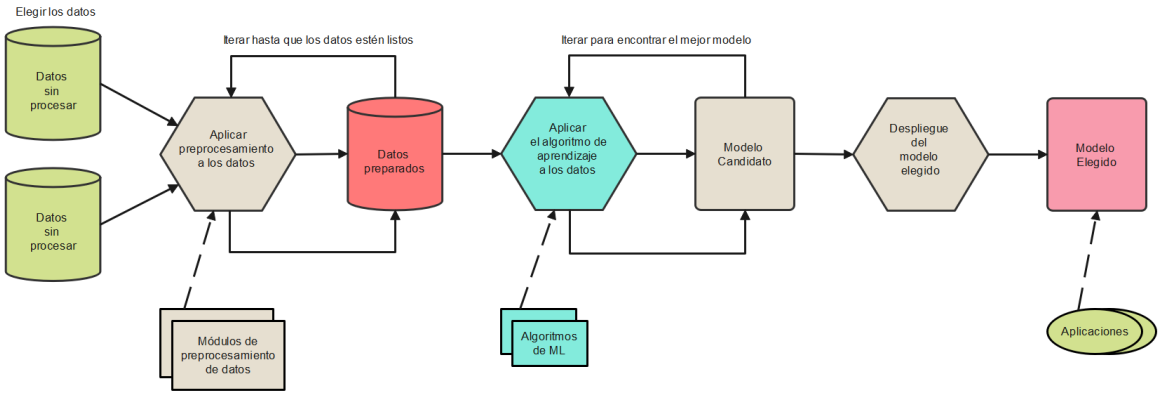


Figura 3.3: Etapas de un proyecto de ML

Es posible observar que cada una de las etapas de un proceso de ML gira en torno a los datos. La necesidad de una gran cantidad de datos de calidad en las fases de entrenamiento y evaluación son la razón principal por la que se generan datos sintéticos para ML.

Redes neuronales

También conocidas como Red Neuronal Artificial o *Artificial Neural Network*s (ANNs) o Red Neuronal Simulada o *Simulated Neural Network*s (SNNs), son un subconjunto de ML y están en el núcleo de los algoritmos de DL. Su nombre y estructura se inspiran en el cerebro humano, e imitan la forma en la que las neuronas biológicas se señalan entre sí [40]. Las redes neuronales son un modelo matemático en el que se basan potentes algoritmos de ML.

Desde un punto de vista más arquitectónico, una red neuronal es un conjunto de neuronas estructuradas en capas totalmente interconectadas. Cada unidad o nodo (neurona artificial), se conecta a otras unidades a través de arcos dirigidos. Cada arco (i,j) sirve para propagar la salida de la unidad i (notada a_i) que sirve como una de las entradas para la unidad j. Cabe mencionar que las entradas y salidas son números. Cada arco (i,j) tiene asociado un peso numérico $w_{i,j}$ que determina la fuerza y el signo de la conexión. Cada unidad calcula su salida en función de las entradas que recibe y la salida de cada unidad sirve, a su vez, como una de las entradas de otras neuronas. La red recibe una serie de entradas externas (unidades de entrada) y devuelve al exterior la salida de algunas de sus neuronas, llamadas unidades de salida; en este sentido se comporta como una función matemática.

La salida de cada unidad se calcula como indica la Ecuación 3.1.

$$a_j = g(\sum_{i=0}^{n} w_{i,j} \cdot a_i) \tag{3.1}$$

En la Ecuación 3.1, g es una función de activación. La función de activación g introduce cierta componente no lineal y aumenta la expresividad del modelo. Cabe mencionar que el sumatorio se hace sobre todas las unidades i que envían su salida a la unidad j, excepto para $i = 0$, que se considera una entrada ficticia, $a_0 = 1$, y un peso $w_{0,j}$ denominado *umbral* o *bias*. Intuitivamente, el umbral $w_{0,j}$ de cada unidad se interpreta como el opuesto de una cantidad cuya entrada debe superar.

En una red neuronal, se distinguen 3 tipos de capas: capa de entrada, capas ocultas (intermedias) y capa de salida. La capa de entrada recoge simplemente la entrada. Para unidades de la misma capa, la función de activación usada es la misma; entre capas distintas puede variar. La función de activación de la capa de salida varía en función del uso que se le quiera dar a la red [41]. Toda esta arquitectura puede observarse en el ejemplo de la Figura 3.4, donde las circunferencias representan cada unidad o nodo (neurona) y se colorean del mismo color las neuronas que pertenecen a la misma capa.

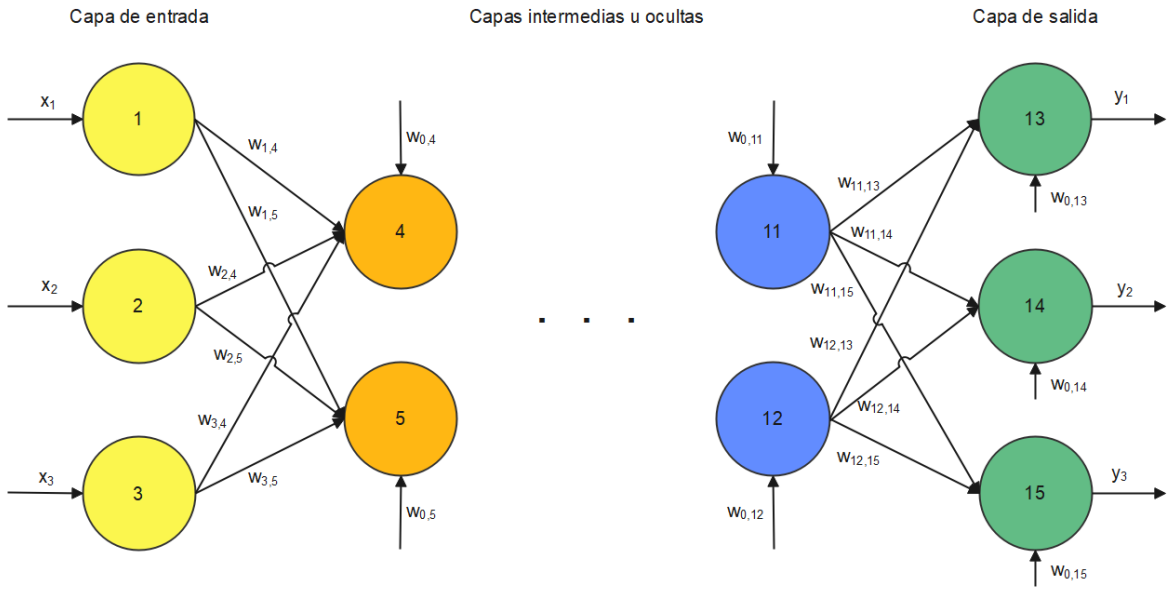


Figura 3.4: Esquema de una red neuronal

Aprendizaje profundo

En la actualidad, dentro de la IA destaca la disciplina del DL, una rama del ML que es extremadamente efectiva en el aprendizaje de patrones. Trabaja con algoritmos que extraen el conocimiento significativo de los datos mediante una jerarquía de múltiples capas que tratan de imitar las redes neuronales del cerebro. Cada capa transforma los datos de entrada en representaciones más abstractas que se van combinando según se profundiza en la red [42]. A diferencia de los algoritmos tradicionales de ML, muchos de los cuales tienen una capacidad finita de aprendizaje, independientemente de cuántos datos adquieran, los sistemas de DL pueden mejorar su rendimiento al poder acceder a un mayor número de datos o, lo que es lo mismo, hacer que la máquina tenga más experiencia. Una vez que las máquinas han conseguido suficiente experiencia mediante DL, pueden ponerse a trabajar para realizar tareas específicas como conducir un coche, detectar hierbajos en un campo de cultivo, detectar enfermedades, inspeccionar maquinaria para identificar errores, etc. [43]

3.1.2. Modelos de ML

En ML, un hiperparámetro es un parámetro cuyo valor se utiliza para controlar el proceso de aprendizaje. Por el contrario, los valores de otros parámetros (típicamente pesos de nodo) se derivan a través del entrenamiento. Los hiperparámetros se pueden clasificar como hiperparámetros de modelo o hiperparámetros de algoritmo. Los primeros, como su propio nombre indica, se refieren a la tarea de selección del modelo y no se pueden inferir al ajustar la máquina al conjunto de entrenamiento. Un ejemplo de un hiperparámetro de modelo es la topología y el tamaño de una red neuronal. Los segundos, que en principio no tienen influencia en el rendimiento del modelo, afectan la velocidad y la calidad del proceso de aprendizaje. Ejemplos de hiperparámetros de algoritmos son la tasa de aprendizaje y el tamaño del lote, así como el tamaño del mini-lote.

El tamaño del lote puede referirse a la muestra de datos completa donde el tamaño del mini-lote sería un conjunto de muestras más pequeño. Los diferentes algoritmos de entrenamiento de modelos requieren diferentes hiperparámetros; por ejemplo, algunos algoritmos simples no requieren ninguno. Dados estos hiperparámetros, el algoritmo de entrenamiento aprende los parámetros de los datos [44].

A continuación, se explican los diferentes algoritmos de ML que han sido utilizados hasta el momento en ingeniería para generar datos sintéticos. Cabe mencionar que existen muchos más modelos de ML pero los dedicados a este propósito son regresión lineal, árbol de decisión, bosque aleatorio, máquina de vectores de soporte, autocodificador variacional y red generativa adversaria.

Regresión lineal En estadística, la regresión lineal es una aproximación para modelar la relación entre una variable escalar dependiente Y y una (regresión lineal simple) o más variables (regresión lineal múltiple) explicativas nombradas con X.

La regresión lineal (*linear regression*) es un algoritmo de aprendizaje supervisado que se utiliza en ML y en estadística. En su versión más sencilla, "dibuja una recta" [1] que indica la tendencia de un conjunto de datos continuos (si fueran discretos, se utilizaría regresión logística).

Este algoritmo aprende por sí mismo a obtener automáticamente esa "recta" con la tendencia de predicción. Para ello, se mide el error con respecto a los puntos de entrada y el valor Y de salida real. El algoritmo debe minimizar el coste de una función de error cuadrático y esos coeficientes corresponden con la recta óptima. Hay diversos métodos para conseguir minimizar el coste; lo más común es utilizar una versión vectorial y la llamada ecuación normal, que da un resultado directo [46].

Árbol de decisión Un árbol de decisión (*decision tree*) es un algoritmo de ML supervisado no paramétrico, que se utiliza tanto para tareas de clasificación como de regresión. Tiene una estructura jerárquica de árbol, que consiste en un nodo raíz, ramas, nodos internos y nodos hoja.

Como se puede ver en la Figura 3.5, un árbol de decisión comienza con un nodo raíz, que no tiene ninguna rama entrante. Las ramas salientes del nodo raíz alimentan a los nodos internos, también conocidos como nodos de decisión. Ambos tipos de nodos realizan evaluaciones sobre la base de las características disponibles para formar subconjuntos homogéneos, que se denotan por nodos hoja o nodos terminales. Los nodos hoja representan todos los resultados posibles dentro del conjunto de datos. Este tipo de estructura de diagrama de flujo también crea una representación fácil para la toma de decisiones, lo que permite a los diferentes grupos de una organización comprender mejor por qué se tomó una decisión.

[1] cuando se habla de "recta" es en el caso particular de regresión lineal simple. También podemos usar otras funciones distintas de la lineal para modelar la relación entre X e Y; por ejemplo, un polinomio [45].

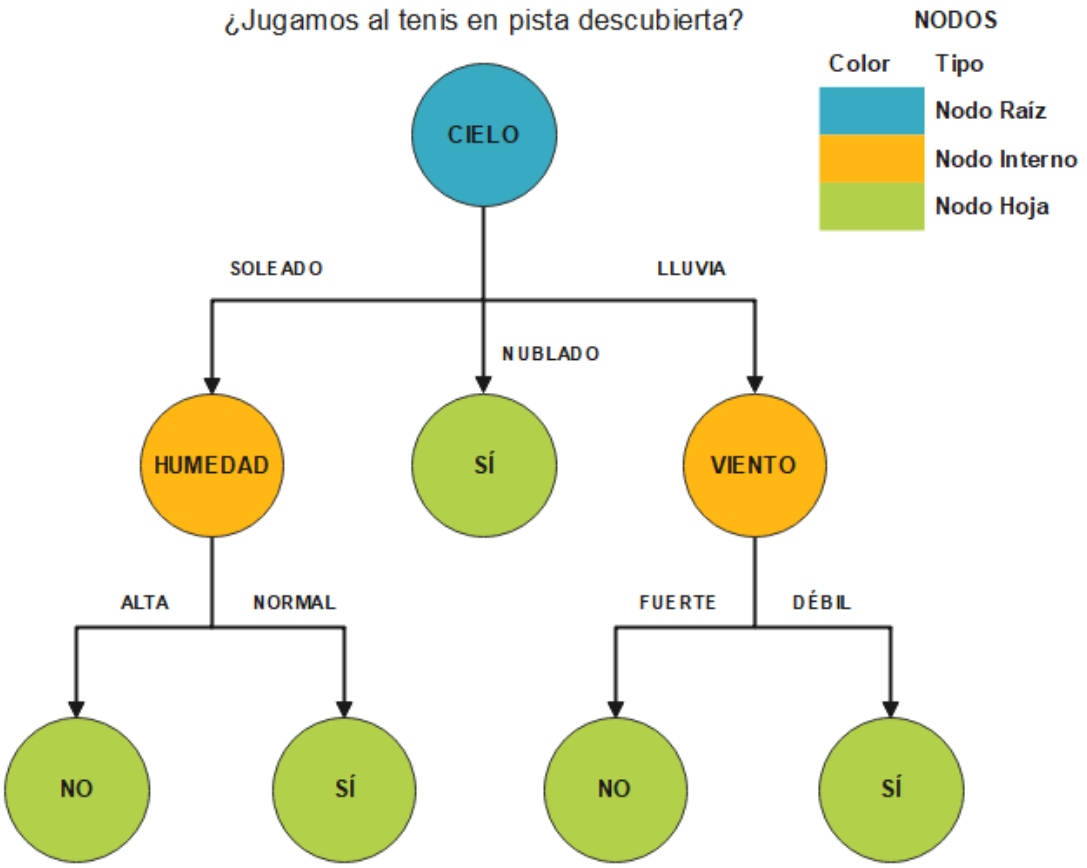


Figura 3.5: Árbol de decisión

El aprendizaje del árbol de decisión realiza una búsqueda codiciosa para identificar los puntos de división óptimos dentro del árbol mediante una estrategia de divide y vencerás. Este proceso de división se repite de manera recursiva de arriba hacia abajo hasta que todos o la mayoría de los registros se hayan clasificado bajo etiquetas de clase específicas. El hecho de que todos los puntos de datos se clasifiquen o no como conjuntos homogéneos depende en gran medida de la complejidad del árbol de decisión. En los árboles más pequeños, es más fácil alcanzar nodos de hoja pura, es decir, clasificar datos individuales en una sola clase. Sin embargo, a medida que un árbol crece en tamaño, se vuelve cada vez más difícil mantener esta pureza, y generalmente muy pocos datos finalizan dentro de un subárbol dado. Cuando esto ocurre, se conoce como fragmentación de datos y, a menudo, puede conducir a un sobreajuste. Como resultado, en consistencia con el principio de parsimonia en la navaja de Occam [2], son preferibles los árboles pequeños. Para reducir la complejidad y evitar el sobreajuste, generalmente se emplea la poda [3]. El ajuste del modelo se puede evaluar a través del proceso de validación cruzada [4] [47].

[2]Este principio dice así: "Las entidades no deben multiplicarse más allá de la necesidad". Dicho de otra manera, los árboles de decisión deben agregar complejidad solo si es necesario, ya que la explicación más simple es a menudo la mejor.

[3]Proceso que elimina las ramas que se dividen en características con poca importancia.

[4]Técnica que consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones.

Bosque aleatorio El bosque aleatorio (*random forest*) es un algoritmo de ML de uso común que combina la salida de múltiples árboles de decisión para alcanzar un solo resultado. Su facilidad de uso y flexibilidad han impulsado su adopción, ya que maneja problemas de clasificación y regresión.

Los árboles de decisión son algoritmos comunes de aprendizaje supervisado, pero pueden ser propensos a problemas, como el sesgo y el sobreajuste. Sin embargo, cuando varios árboles de decisión forman un conjunto en el algoritmo de bosque aleatorio, predicen resultados más precisos, particularmente cuando los árboles individuales no están correlacionados entre sí [48].

Los métodos de aprendizaje conjunto (*ensembles*), como el bosque aleatorio, se componen de un conjunto de clasificadores (en este caso, árboles de decisión) y sus predicciones se agregan para identificar el resultado más popular. Los métodos de conjunto más conocidos son el embolsado (*bagging* o *bootstrap aggregating*) y el *boosting*. En el método de embolsado, se selecciona una muestra aleatoria de datos en un conjunto de entrenamiento con reemplazo, lo que significa que los puntos de datos individuales se pueden elegir más de una vez. Después de que se generen varias muestras de datos, estos modelos se entrenan de forma independiente; dependiendo del tipo de tarea, es decir, regresión o clasificación, el promedio o la mayoría de esas predicciones producen una estimación más precisa. Este enfoque se usa comúnmente para reducir la varianza dentro de un conjunto de datos ruidoso. Por su parte, el *boosting* es un método de aprendizaje conjunto que combina un conjunto de estudiantes débiles en un alumno fuerte para minimizar los errores de entrenamiento. En el impulso, se selecciona una muestra aleatoria de datos, se ajusta a un modelo y luego se entrena secuencialmente, es decir, cada modelo intenta compensar las debilidades de su predecesor. Con cada iteración, las reglas débiles de cada clasificador individual se combinan para formar una regla de predicción fuerte [49].

El algoritmo de bosque aleatorio es una extensión del método de embolsado, ya que utiliza tanto el embolsado como la aleatoriedad de características para crear un bosque no correlacionado de árboles de decisión. La aleatoriedad de características, también conocida como embolsado de características o "el método de subespacio aleatorio" genera un subconjunto aleatorio de características, lo que garantiza una baja correlación entre los árboles de decisión. Esta es una diferencia clave entre los árboles de decisión y los bosques aleatorios. Mientras que los árboles de decisión consideran todas las posibles divisiones de características, los bosques aleatorios solo seleccionan un subconjunto de esas características. Al tener en cuenta toda la variabilidad potencial en los datos, hace posible reducir el riesgo de sobreajuste, sesgo y varianza general, lo que resulta en predicciones más precisas [50].

Máquina de vectores de soporte La Máquina de Vectores de Soporte o *Support-Vector Machine* (SVM) es una técnica de clasificación y regresión que aprovecha al máximo la precisión de las predicciones de un modelo sin ajustar excesivamente los datos de entrenamiento. SVM es ideal para analizar datos con un gran número de campos de predictores (por ejemplo, miles).

SVM funciona correlacionando datos a un espacio de características de grandes dimensiones de forma que los puntos de datos se puedan categorizar, incluso si los datos no se pueden separar linealmente de otro modo. Se detecta un separador entre las categorías y los datos se transforman de forma que el separador se puede extraer como un hiperplano. Tras ello, las características de los nuevos datos se pueden utilizar para predecir el grupo al que pertenece el nuevo registro.

SVM tiene aplicaciones en multitud de disciplinas, incluyendo el Gestión de Relación con los Clientes o *Customer Relationship Management* (CRM), el reconocimiento facial y de otras imágenes, la bioinformática, la extracción de conceptos de minería de texto, la detección de

intrusiones, la predicción de estructuras de proteínas y el reconocimiento de la voz [51].

Autocodificador variacional Se llama codificador al proceso que produce la representación de nuevas características a partir de la representación de características antiguas (por selección o por extracción) y decodificador al proceso inverso. Este proceso se puede observar en el diagrama de flujo de la Figura 3.6. La reducción de dimensionalidad puede interpretarse como compresión de datos donde el codificador comprime los datos (desde el espacio inicial hasta el espacio codificado, también llamado espacio latente) mientras que el decodificador los descomprime. Para un conjunto dado de posibles codificadores y decodificadores, se busca el par que mantiene el máximo de información al codificar y, por lo tanto, tiene el mínimo de error de reconstrucción al decodificar. En resumen, un Autocodificador Variacional o *Variational Autoencoder* (VAE) es un autocodificador cuya distribución de codificaciones se regulariza durante el entrenamiento para garantizar que su espacio latente tenga buenas propiedades que permitan generar algunos datos nuevos. El término "variacional" proviene de la estrecha relación que existe entre la regularización y el método de inferencia variacional en estadística [52].

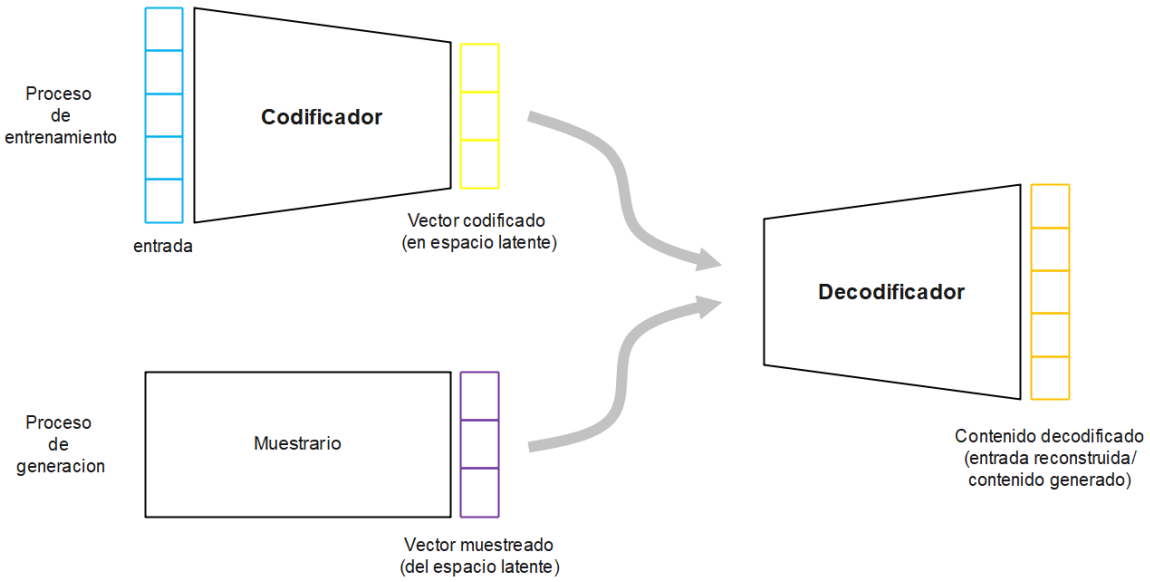


Figura 3.6: Autocodificador variacional

Red generativa adversaria En el modelo de Red Generativa Adversaria o *Generative Adversarial Network* (GAN), dos redes neuronales antagónicas, generador y discriminador, entrenan el modelo iterativamente. El generador toma datos de muestra aleatorios y genera un conjunto de datos sintético. El discriminador compara los datos generados sintéticamente con un conjunto de datos real basado en las condiciones que se establecen anteriormente [53]. Este proceso se puede observar en el diagrama de flujo de la Figura 3.7.

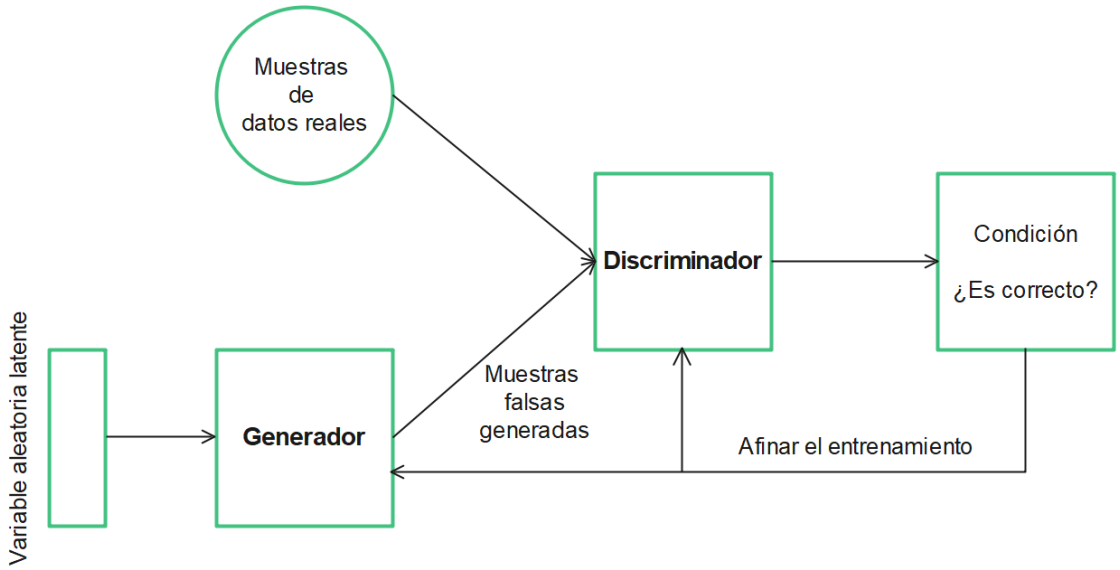


Figura 3.7: Red generativa adversaria

3.2. Datos sintéticos

Los datos son el núcleo de las actividades actuales de ciencia de datos e inteligencia de negocio. Como se ha mencionado anteriormente, existen múltiples escenarios en la organización en los que los datos no pueden circular dentro de departamentos, subsidiarias o socios. En estos casos, los datos sintéticos se pueden utilizar como un reemplazo directo para cualquier tipo de comportamiento, análisis predictivo o transaccional [3].

Hay varias razones detrás de la necesidad de datos sintéticos; sin embargo, aquí se destacan solo tres de ellas. En primer lugar, puede ser una cuestión de **disponibilidad**; una organización o equipo no tiene los datos suficientes. Para las organizaciones más grandes, las infraestructuras heredadas y los sistemas de datos en silos también suelen ser una causa de falta de disponibilidad de datos. En el panorama regulatorio actual de la protección de datos, también puede ser una cuestión de **cumplimiento legal**; los datos existen, pero su procesamiento está estrictamente regulado. Por ejemplo, el Reglamento General de Protección de Datos o *General Data Protection Regulation* (GDPR) [54]) prohíbe los usos que no fueron explícitamente consentidos cuando la organización recopiló los datos. Las preocupaciones de seguridad también pueden evitar que los datos fluyan dentro de una organización; por ejemplo, porque la información es demasiado sensible para ser migrada a una infraestructura en la nube. Los procesos de gobierno también pueden ralentizar o limitar el acceso a los datos por razones similares. Finalmente, puede reducirse a una cuestión de **coste**; un determinado activo de datos puede ser demasiado caro de comprar o puede llevar mucho tiempo acceder a él y prepararlo [32].

Tanto los encargados de tomar decisiones dentro de una organización, como los investigadores, necesitan datos para validar sus hipótesis. Para tener análisis fieles, los conjuntos de datos publicados deben conservar la utilidad del conjunto de datos original; pero, como se cita en el párrafo anterior, la privacidad de los propietarios de los datos es una preocupación igualmente

importante. Para mitigar el riesgo de violación de la confidencialidad, las agencias emplean diferentes técnicas, como reordenar o recodificar variables sensibles o barajar valores entre diferentes registros. A pesar de estos esfuerzos de las agencias, se tienen ejemplos de violaciones de confidencialidad en conjuntos de datos anónimos. Si se mantiene la privacidad de los datos como el único objetivo, entonces la utilidad de los datos se ve muy comprometida. Por lo tanto, existe la necesidad de una forma de generar conjuntos de datos que puedan ponerse a disposición del público, con un riesgo mínimo de divulgación de datos y una utilidad máxima [55].

Para determinar el mejor método para crear datos sintéticos, es importante considerar primero qué tipo de datos sintéticos se desea tener. Atendiendo a la naturaleza de los datos, hay dos categorías amplias para elegir. Éstas se explican a continuación:

- *Discretos*. Los datos discretos pueden ser numéricos, como números de manzanas; pero también pueden ser categóricos, como rojo o azul, masculino o femenino, bueno o malo, etc.

- *Continuos*. Los datos continuos no están restringidos a valores separados definidos, sino que pueden ocupar cualquier valor en un rango continuo.

Atendiendo a la proporción de originalidad en los datos de salida, hay tres categorías amplias para elegir [32] [3] [56]. Éstas se explican a continuación y se pueden observar en la Figura 3.8:

- *Totalmente sintético*. Estos datos no contienen ningún dato original; esto significa que la identificación de cualquier unidad individual es casi imposible. A menudo se usan cuando la privacidad impide el uso de los datos originales.

- *Parcialmente sintético*. Solo los datos que son confidenciales se reemplazan con datos sintéticos. Esto conduce a una disminución de la dependencia del modelo, pero significa que es posible que haya cierta divulgación, debido a los valores verdaderos que permanecen dentro del conjunto de datos. Estos datos ayudan a complementar los conjuntos de datos existentes cuando falta cierta información o la cantidad de datos no es suficiente para una aplicación determinada.

- *Híbrido sintético*. Se usan en los casos en que solo existe una parte de los datos reales. Los datos sintéticos híbridos se derivan de datos reales y sintéticos. En primer lugar, se investiga la distribución subyacente de los datos originales. Seguidamente, para cada registro de datos reales se elige un registro cercano en los datos sintéticos. Al final, los dos se unen para generar datos híbridos. Los datos parciales sintéticos, a diferencia de estos híbridos, pueden contener registros totalmente reales.

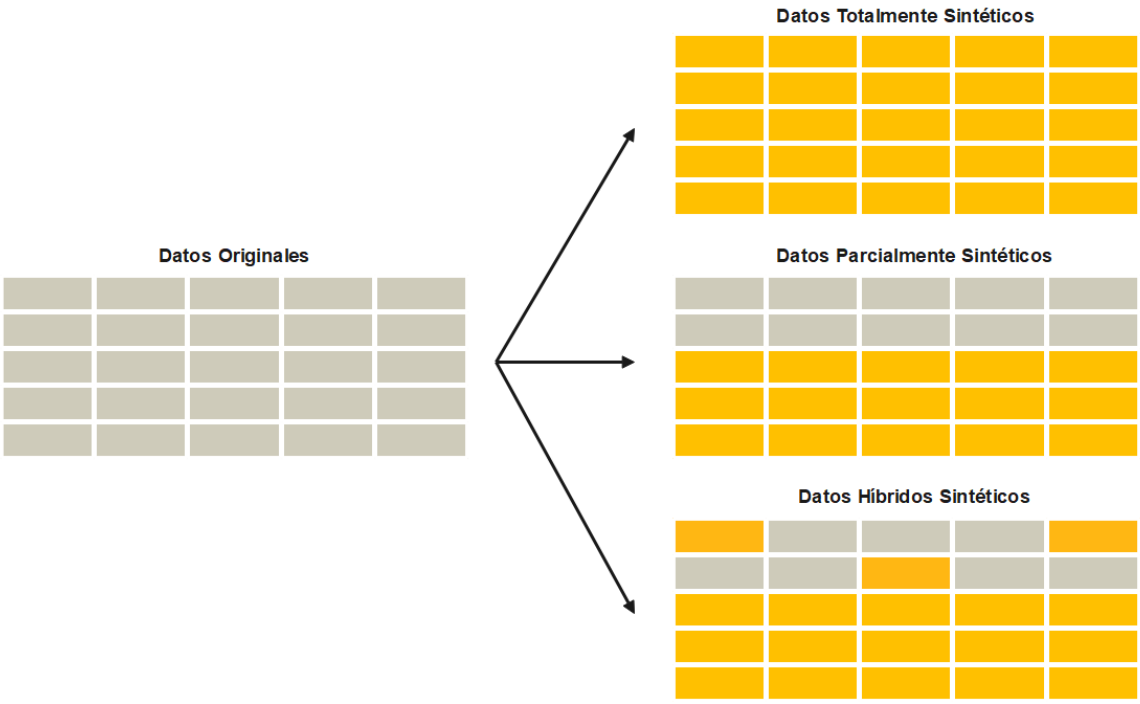


Figura 3.8: Tipos de datos sintéticos (tabulares) atendiendo a la proporción de originalidad en los datos de salida

Ahora bien, hay varios tipos de datos sintéticos que sirven para diferentes propósitos. Los datos sintéticos según su estructura pueden ser [32] [3]:

- *Texto sintético.* Es texto generado artificialmente. Debido a las complejidades de los idiomas, generar texto sintético realista siempre ha sido un desafío. Sin embargo, el surgimiento de nuevos modelos de ML condujo a la concepción de sistemas de generación de lenguaje natural de notable rendimiento. Se crea y entrena un modelo para generar texto.

- *Multimedia sintético.* Los datos sintéticos también pueden ser vídeo, imagen o sonido sintéticos. Se puede representar artificialmente multimedia con propiedades lo suficientemente cercanas a los datos de la vida real. Esta similitud permite utilizar los medios sintéticos como un reemplazo directo de los datos originales. Además de los motivos mencionados anteriormente, se pueden utilizar datos sintéticos para aumentar el tamaño y la diversidad de los conjuntos de datos al entrenar sistemas de reconocimiento de imágenes.

- *Datos sintéticos tabulares.* Se refieren a los datos generados artificialmente que imitan los datos de la vida real almacenados en tablas. Estos datos se estructuran en filas y columnas. Podría ser cualquier cosa, desde una base de datos de pacientes hasta información de comportamiento analítico de usuarios o registros financieros. El sistema *software* fruto de este proyecto se centra solo en la generación de este tipo de datos sintéticos. Un ejemplo de la colección de datos sintéticos tabulares que va a generar el sistema se puede observar en la Tabla 3.1. El significado de cada una de las columnas se explica en la Tabla 4.10 de la sección 4.3.3.

Día	Susceptibles	Expuestos	Infectados	Recuperados
0	99999	0	1	0
1	99567	403	30	0
2	99385	507	108	0
3	99311	487	202	0

Tabla 3.1: Ejemplo de datos sintéticos tabulares

Aunque los datos sintéticos tienen varios beneficios que pueden facilitar los proyectos de ciencia de datos para las organizaciones, también tienen una serie de limitaciones que se van a comentar a continuación [32].

- Es posible que falten valores atípicos; los datos sintéticos solo pueden imitar los datos del mundo real, no es una réplica exacta de ellos. Por lo tanto, es posible que los datos sintéticos no cubran algunos valores atípicos que tienen los datos originales. Sin embargo, los valores atípicos en los datos pueden ser más importantes que los puntos de datos regulares.
- La calidad del modelo depende de la fuente de datos, esto es, la calidad de los datos sintéticos está altamente correlacionada con la calidad de los datos de entrada y el modelo de generación de datos. Los datos sintéticos pueden reflejar los sesgos en los datos de origen.
- La aceptación del usuario es más desafiante, pues los datos sintéticos son un concepto emergente y pueden no ser aceptados como válidos por los usuarios que no han sido testigos de sus beneficios antes.
- La generación de datos sintéticos requiere tiempo y esfuerzo y, aunque son más fáciles de crear que los datos reales, tampoco son gratuitos.
- Es necesario un control de salida, especialmente en conjuntos de datos complejos. En este caso, la mejor manera de garantizar que la salida sea precisa es comparando datos sintéticos con datos auténticos o datos anotados por humanos. Esto se debe a que podría haber incoherencias en los datos sintéticos al intentar replicar complejidades dentro de los conjuntos de datos originales.

Llega ahora el tema central de este proyecto; existen tres estrategias de referencia para la generación de datos sintéticos. Éstas se describen en las siguientes subsecciones y se derivan de las fuentes [57] y [58].

3.2.1. Procesos estocásticos

Se trata de una generación aleatoria de datos. En consecuencia, las características, relaciones y patrones estadísticos que se encuentran en los datos originales no se conservan, capturan, ni reproducen en los datos ficticios generados. Por lo tanto, la representatividad de los datos simulados es mínima en comparación con los datos originales. En este caso, la forma importa más que el contenido. Si se conoce la estructura de los datos sintéticos deseados y la distribución de los datos es irrelevante, el proceso estocástico es un método perfecto.

La aplicabilidad de estos procesos se limita a los casos en que el contenido de los datos sintéticos es irrelevante y el ruido aleatorio es lo suficientemente bueno en lugar de los datos reales. Ejemplos de tales aplicaciones serían los sistemas de pruebas de estrés, donde se genera una gran cantidad de datos aleatorios sobre la marcha para evaluar cómo se comportan los sistemas bajo un uso intensivo. Se usa también para reemplazar Información de Identificación Personal o *Personally Identifiable Information*s (PIIs) o cuando no se dispone de suficiente tiempo y energía para definir reglas.

3.2.2. Generación de datos basada en reglas

La desventaja obvia de los datos generados con procesos estocásticos son sus casos de uso limitados, ya que los datos son aleatorios y no contienen información real. Los sistemas de generación de datos basados en reglas lo mejoran mediante el uso de datos generados siguiendo reglas específicas definidas por humanos, es decir, los datos sintéticos son generados en base a un conjunto predefinido de reglas. Ejemplos de esas reglas de un determinado atributo numérico podrían ser el valor mínimo, el valor máximo o el valor promedio. Se puede predefinir de esta forma cualquier característica, relación o patrón estadístico de los datos originales que se desee reproducir en los nuevos datos. La complejidad de esas reglas puede variar desde muy simples, teniendo en cuenta solo el tipo de datos deseado en una columna (es decir, si una columna contiene datos numéricos o categóricos), hasta reglas más sofisticadas, que definen las relaciones entre varias columnas y eventos. La cantidad de trabajo humano y la experiencia necesaria, así como la información contenida en los datos generados, dependen completamente de las reglas definidas.

Por lo tanto, los métodos basados en reglas tienen tres desafíos:

- Escalabilidad. Los conjuntos de datos que contienen muchas columnas interdependientes diferentes en una configuración de varias tablas necesitan fácilmente cientos de reglas complejas y entrelazadas. Agregar reglas adicionales se vuelve cada vez más difícil, lo que prácticamente limita la complejidad máxima de los datos que se pueden modelar.

- Sesgo. Dado que las reglas son definidas por expertos humanos, el sesgo de esos expertos se refleja en las reglas y, por lo tanto, está presente en los datos generados. Algunas columnas de la tabla pueden reflejar una lógica empresarial claramente definida, donde el sesgo es parte de la política acordada, pero otras (por ejemplo, el comportamiento del cliente o el historial del paciente) pueden ser más susceptibles al sesgo humano inconsciente.

- Derivabilidad. Los datos relacionados con el mundo real cambian continuamente, por lo que las reglas deben modificarse para reflejar ese cambio. Los sistemas complejos basados en normas necesitan una gestión eficaz del cambio que rija las condiciones y decida qué normas deben adaptarse para reflejar los cambios en el contexto de su aplicación.

Hacer frente a estos desafíos puede ser muy difícil y, en muchos casos, son factores decisivos. Específicamente, la escalabilidad y la derivabilidad evitan que los sistemas basados en reglas se utilicen en aplicaciones que requieren flexibilidad y soporte para requisitos de datos cambiantes, limitando efectivamente su aplicabilidad a casos de uso donde el alcance y los requisitos de datos son exactamente conocidos y no cambiarán. Pero si estos desafíos se cumplen con éxito, un sistema basado en reglas puede ser una opción lo suficientemente buena para probar en aplicaciones, que van desde la generación de datos tabulares hasta contenido multimedia.

Dentro de este grupo se encuentran las técnicas de extracción de números de una distribución, ya que estos números pueden considerarse una regla más. Este método funciona observando distribuciones estadísticas reales y reproduciendo datos falsos. Esto también puede incluir la creación de modelos generativos. Existen dos técnicas de extracción de números de una distribución que se listan y describen a continuación [53].

- Generación según distribución. Para los casos en que no existen datos reales pero el analista de datos tiene una comprensión integral de cómo se vería la distribución del conjunto de datos. El analista puede generar una muestra aleatoria de cualquier distribución, como Normal, Exponencial, Chi-cuadrado, t, lognormal y Uniforme. En esta técnica, la utilidad de los datos sintéticos varía según el grado de conocimiento del analista sobre un entorno de datos específico.

- Ajuste de datos reales a una distribución conocida. Si hay datos reales, entonces las empresas pueden generar datos sintéticos determinando las distribuciones que mejor se ajustan a los datos reales dados.

3.2.3. Modelos de ML

La complejidad de los datos que se pueden aprender mediante un modelo de este tipo está, principalmente, limitada por los datos disponibles y la capacidad del modelo (es decir, la arquitectura del modelo y los hiperparámetros). Si los requisitos de datos cambian, no se necesitan ajustes significativos, simplemente se debe entrenar un nuevo modelo sobre los datos reales.

Debido al poder de los modelos de ML (que imitan los datos de entrenamiento), se deben abordar tres nuevos desafíos únicos para este enfoque:

- Similitud de datos. El éxito de replicar la información contenida en los datos originales depende de la complejidad de los datos y de la capacidad del modelo que uno elija utilizar. Para obtener los mejores resultados, se debe prestar especial atención a probar y documentar la similitud de los datos sintéticos en comparación con los datos originales utilizados para entrenar el modelo. La métrica de similitud es específica de la aplicación, pero los métodos de la estadística descriptiva se pueden utilizar para analizar la distribución univariada y multivariante, así como la correlación entre las características de los datos sintéticos y de entrenamiento.

- Privacidad. Dado que muchos modelos de ML son propensos al sobreajuste, es necesario tener especial precaución para evitar la memorización de ejemplos de entrenamiento. Esto es particularmente importante si los datos de entrenamiento son sensibles a la privacidad y deben protegerse.

- Reglas de negocio. Un modelo bien entrenado puede aprender la mayoría de las reglas y retener las relaciones contenidas en los datos de entrenamiento. Las reglas que aún no se alcancen se pueden hacer cumplir simplemente filtrando los pocos registros no válidos como parte del procesamiento posterior.

Una vez que se han cumplido estos desafíos, las aplicaciones de los datos sintéticos generados por IA son casi ilimitadas. De hecho, surgen dos oportunidades únicas con el uso de estos modelos; una es el uso de datos sintéticos en lugar de los datos originales a los que no se puede acceder

debido a razones legales y de privacidad, y la segunda es el uso de datos sintéticos dentro de una organización para reducir el tiempo de desarrollo de los modelos de ML.

Por ejemplo, los datos sintéticos están desempeñando un papel clave en el desbloqueo de datos originales protegidos por razones de privacidad en el mundo de las finanzas [59] [60]. Aquí, los datos sintéticos se utilizan, entre otras cosas, para mejorar la detección de fraudes, ya que contienen las propiedades estadísticas necesarias para mejorar los sistemas de detección sin exponer la privacidad de las personas. Además, compartir datos entre departamentos e incluso fronteras de países se convierte en un proceso sin fisuras cuando se utiliza una herramienta generativa profunda de alta calidad, precisa y compatible con la privacidad.

Otro ejemplo es el uso de datos sintéticos por parte de las unidades de ciencia de datos, ML e inteligencia de negocio dentro de una organización. En la mayoría de los entornos de trabajo, el acceso a los datos está estrictamente regulado, lo que resulta en procesos que consumen mucho tiempo. Trabajar con datos sintéticos permite construir modelos mucho más rápido y reducir el tiempo de comercialización del modelo. Se ha demostrado en un estudio reciente [61] que los modelos entrenados con datos sintéticos logran resultados comparables y, en algunos casos, incluso superan a los modelos entrenados con datos originales.

Cabe destacar que técnicas como el VAE) o los modelos GAN mejoran la utilidad de los datos al alimentar los modelos con más datos y entran dentro de los llamados modelos generativos profundos. Los modelos generativos profundos son una clase de modelos estadísticos que aprenden la distribución de los datos de entrenamiento y se pueden utilizar para generar nuevos datos después de ese aprendizaje. Aplicando algoritmos de DL a partir del ML, es posible entrenar un modelo (por ejemplo, una red neuronal artificial) con datos reales para que aprenda la estructura y la información contenida y sea capaz de generar nuevos datos sintéticos. La guía humana que necesita un sistema de este tipo puede ser mínima. En el mejor de los casos, no se necesita interacción humana y el modelo de ML se entrena automáticamente.

3.3. Estado del arte

En esta sección, se comienza comparando cualitativamente los tres grandes grupos de técnicas de generación de datos, algo más genérico, para continuar con la comparación cuantitativa entre técnicas específicas del tercer grupo (modelos de ML). Cabe mencionar que ambas comparaciones están publicadas en la web en [58] y en [55] y [62], respectivamente, por lo que no son originales de este proyecto.

3.3.1. Comparativa cualitativa

No todos los datos sintéticos se crean de la misma manera y, además, su generación hoy en día difiere mucho de lo que era años atrás. A continuación, se van a comparar los diferentes métodos de generación de datos sintéticos, desde las formas más rudimentales hasta los métodos de vanguardia, para ver hasta qué punto ha avanzado la tecnología. En esta comparativa se va a distinguir entre los tres métodos principales comentados en la sección 3.2:

- Procesos estocásticos. Los datos generados imitan únicamente la estructura de los datos reales.
- Generación de datos basada en reglas. Los datos se generan siguiendo reglas específicas definidas por los humanos.

- Modelos de ML. Los datos, generados por un modelo de ML entrenado en datos reales, replicando su estructura y la información que contienen, son ricos y realistas.

Para poder decidir qué técnica se debe utilizar en cada caso, han de tenerse en cuenta distintas métricas de evaluación. La elección del método depende del caso de uso y debe ser evaluada, si es posible, tanto por un experto en la síntesis de datos como por un experto en el dominio, que esté familiarizado con los datos y su uso posterior.

Además de los criterios específicos de casos de uso, se pueden utilizar varios aspectos generales para evaluar y comparar los diferentes métodos disponibles. A continuación, se explican las métricas o dimensiones comparativas que se ponen en práctica en este estudio.

- Computación. Cantidad de cómputo necesaria para generar datos o para construir el modelo.
- Trabajo humano. Cantidad de experiencia y trabajo humano que entra en el proceso de generación.
- Complejidad del sistema. Dificultad para construir el sistema de generación de datos.
- Contenido de la información. Cantidad de información real que está presente en los datos sintéticos. Cuanto más alto mejor, ya que la utilidad de los datos es mayor (no se tienen en cuenta los riesgos de divulgación).

Descripción individual de los trabajos relacionados

Este subapartado provee de un resumen individual de los diferentes trabajos relacionados con el tema principal del proyecto, atendiendo a las dimensiones comparativas propuestas. Este repaso previo a la discusión conjunta de todas ellas se considera necesario para su mejor comprensión.

La generación de datos aleatorios mediante procesos estocásticos tiene necesidades computacionales muy pequeñas; se puede realizar sobre la marcha siempre que se necesiten. La experiencia humana se reduce al mínimo con esta técnica dado que la estructura de los datos sintéticos se puede definir de forma sencilla, o inferir de un conjunto de datos existente. Estos sistemas son los más fáciles de construir; los desafíos durante la implementación derivan del reto de maximizar la cantidad de datos que se pueden generar con recursos determinados. Además, los datos sintéticos generados no contienen información relevante que ponga en peligro la confidencialidad de los datos.

A pesar de que los recursos computacionales necesarios para ejecutar un sistema de generación de datos basado en reglas dependen en su totalidad del número y la complejidad de las reglas, sus necesidades de cálculo se pueden clasificar como menores o moderadas. La cantidad de trabajo humano y experiencia que se necesita para construir dicho sistema es extensa, y mucho más alta que para cualquiera de los otros métodos. El sistema crece con el número y la complejidad de las reglas soportadas. El formato o lenguaje y la interfaz utilizados para describir las reglas pueden contribuir también en la complejidad del sistema que, en general, es alta. La información contenida en los datos generados está limitada únicamente por las reglas que se aplican.

Por último, el entrenamiento de modelos de ML es, en general, de computación intensiva. Este problema puede mitigarse mediante el uso de *hardware* especial (por ejemplo, tarjetas gráficas, TPU, ASIC y otros) y computación en la nube. Un sistema óptimo de ML puede minimizar el trabajo humano; siendo independiente de la aplicación, limita la interacción con el

usuario a seleccionar los datos de entrenamiento utilizados para crear los modelos y a algún post-procesamiento, dependiendo de las cualidades del conjunto de datos y su aplicación. Por otra parte, es el más complejo de los sistemas descritos; la arquitectura de los modelos generativos para datos tabulares y su poder para resolver problemas previamente intactos es un tema de investigación abierto lleno de potenciales futuros. Un modelo de este tipo es capaz de generar datos sintéticos altamente similares a los datos de entrenamiento, maximizando el contenido de la información y superando a los sistemas basados en reglas por un amplio margen. En algunos casos, el resultado es incluso mejor que los datos en bruto.

Toda esta información individual se recoge en la Tabla 3.2 en forma de resumen de los métodos comparados y su rendimiento con las métricas utilizadas para evaluarlos. En esta tabla se utiliza un código de colores en el cual el verde implica la una valoración positiva de una dimensión para una técnica, mientras que la roja es negativa. Se usa el color naranja para una valoración media.

Técnica / Métrica	Procesos estocásticos	Generación basada en reglas	Modelos de ML
Computación			
Trabajo humano			
Complejidad del sistema			
Contenido de la información			
Casos de uso	Pruebas de estrés	Pruebas de estrés, pruebas de *software* sencillas	Desbloqueo de datos privados, análisis avanzados, desarrollo y capacitación de modelos de ML, retención de datos, colaboración en investigación...

Tabla 3.2: Resumen comparación cualitativa

Discusión

Tras discutir las capacidades y los desafíos de los diversos métodos de generación de datos sintéticos, decidir cuál se adapta mejor a los requisitos y casos de uso de cada trabajo en cuestión es una tarea aún ardua y es necesario responder a dos preguntas cerradas. La primera es si se dispone de recursos y experiencia para construir el sistema; la segunda es si los datos sintéticos deben ser realistas y representativos.

Por un lado, las soluciones que hacen uso de procesos estocásticos o sistemas basados en reglas dependen en gran medida de su caso de uso, y casi siempre requieren el desarrollo de un nuevo *software*. Existen bibliotecas de apoyo para tales esfuerzos, pero requieren experiencia, recursos y voluntad para ser mantenidas. Por otro lado, los procesos estocásticos están fuera de discusión cuando los datos sintetizados tienen que ser realistas. Por su parte, los sistemas basados en reglas solo tienen sentido si está claro cómo deben ser los datos y esa descripción se puede escribir en

código.

Si el desarrollo "casero" no es una opción y los datos sintéticos deben ser lo más realistas y representativos posible, el uso de sistemas habilitados para ML es la mejor opción.

3.3.2. Comparativa cuantitativa

En primer lugar, se definen tres conceptos nuevos, variable categórica, variable cualitativa e Imputación Múltiple (IM), que van a aparecer a lo largo de la subsección. Se continúa describiendo tanto las dimensiones comparativas como la metodología de trabajo usadas para realizar la comparación experimental entre algoritmos de ML. Finalmente, se entra de lleno en esta comparación.

En estadística, una variable categórica es una variable que puede tomar uno de un número limitado, y por lo general fijo, de posibles valores, asignando a cada unidad individual u otro tipo observación a un grupo en particular o categoría nominal sobre la base de alguna característica cualitativa. En informática y algunas ramas de las matemáticas, las variables categóricas se conocen como enumeraciones o tipos enumerados. Comúnmente, cada uno de los posibles valores de una variable categórica se conoce como un nivel. La distribución de probabilidad asociada con una variable categórica se llama una distribución categórica [63]. A una variable se le denomina variable cualitativa ordinal o variable cuasicuantitativa cuando dicha variable puede tomar distintos valores ordenados siguiendo una escala establecida, aunque no es necesario que el intervalo entre mediciones sea uniforme, por ejemplo: leve, moderado, fuerte [64].

La IM es una forma de lidiar con el sesgo de los datos faltantes, permitiendo analizar datos incompletos con herramientas regulares de análisis de datos. Es importante saber aquí que un sinónimo de imputar es completar. Con los métodos de imputación singular, se imputan los valores faltantes con la media, la mediana u otros parámetros estadísticos. Sin embargo, el uso de valores únicos conlleva un nivel de incertidumbre sobre qué valores imputar. La IM reduce la incertidumbre sobre los valores faltantes mediante el cálculo de varias opciones diferentes (imputaciones). Se crean varias versiones del mismo conjunto de datos, que luego se combinan para obtener los "mejores" valores [65].

Los conjuntos de datos total y parcialmente sintéticos cierran la brecha entre los temas contrastantes de privacidad y utilidad de los datos. Usan la IM para generar registros sintéticos que preservan las relaciones en la población. Por tanto, los valores de los atributos generados sintéticamente son tratados como valores faltantes que se generan utilizando diferentes herramientas de ML, como árboles de decisión, bosque aleatorio, la máquina de vectores de soporte y otros modelos.

A continuación, se van a evaluar comparativamente las técnicas de generación de datos sintéticos basadas en ML utilizando diferentes sintetizadores de datos: a saber, regresión lineal, árbol de decisión, bosque aleatorio y red neuronal. Se evalúa su efectividad en términos de cuánta utilidad se retiene y el riesgo de divulgación de datos individuales. Para evaluar el riesgo de divulgación, éste se define en función de los supuestos de escenarios de divulgación de datos, es decir, los escenarios en los que un intruso podría explotar los datos publicados para revelar información de un registro del conjunto de datos. Se evalúa también la eficiencia de cada modelo en términos de tiempo requerido para generar conjuntos de datos sintéticos. En resumen, las dimensiones comparativas consideradas para evaluar conjuntamente los diferentes sintetizadores de datos serían la utilidad retenida, el riesgo de divulgación de datos individuales y el tiempo requerido por el método en cuestión para generar datos.

Es posible generar conjuntos de datos completamente sintéticos, en los que los valores de ciertos atributos se reemplazan mediante IM para todos los elementos en el conjunto de datos. Aunque es ventajoso generar valores sintéticamente para todos los puntos de datos, no siempre es una necesidad. Los conjuntos de datos parcialmente sintéticos se obtienen generando sintéticamente sólo los valores de los atributos que son sensibles a la divulgación pública.

Se utilizan varios sintetizadores de datos como el árbol de decisión, el bosque aleatorio o la máquina de vectores de soporte para generar datos total y parcialmente sintéticos. Recientemente, se ha creado un paquete R, *synthpop* [66], que proporciona funcionalidades básicas para generar conjuntos de datos sintéticos y realizar una evaluación estadística.

Tras esta introducción, es posible pasar a explicar la metodología que se utiliza para llevar a cabo la comparación. Se considera una población, P, con un conjunto de características. Un imputador realiza un análisis cualitativo de las características (atributos) frente al riesgo de divulgación y divide el conjunto de características en dos subconjuntos separados: X, un conjunto de características de identificación, e Y, un conjunto de características sensibles. Y contiene datos confidenciales sobre los registros de la población, mientras que X contiene los datos que pueden publicarse o los datos que están disponibles para las personas de algunas fuentes alternativas. Por otro lado, los datos se dividen en dos conjuntos: P_{obs} y P_{nobs}. P_{nobs} es un conjunto de datos reservado que nunca se publica y las muestras extraídas de P_{obs} se publican como conjuntos de datos para uso público. El problema es concebir un mecanismo para liberar muestras de P_{obs} de modo que no se divulgue información confidencial.

Por un lado, se propone el uso de IM para generar conjuntos de datos completamente sintéticos. Dadas las características de identificación, las técnicas de IM utilizan datos reservados para ajustar una distribución posterior sobre las características sensibles. Tras ello, completan los valores faltantes generando muestras a partir de esta distribución posterior. Por otro lado, ocurre muchas veces que diferentes registros tienen diferente riesgo de divulgación dependiendo de su contenido. Los registros con un cierto rango de valores para características sensibles tienen un riesgo muy alto de divulgación en comparación con otros valores de características sensibles. Por lo tanto, no siempre es necesario generar valores sintéticos de características sensibles para todos los registros en P_{obs}. Se propone entonces una generación de datos parcialmente sintética que utiliza la IM que genera valores sintéticos solo para aquellos registros en P_{obs} que tienen un riesgo muy alto de divulgación para características en Y.

También es posible ampliar la idea de la IM al uso de modelos de ML para generar los datos de forma sintética. El enfoque general de usar el modelo de ML es entrenar el modelo en los registros en P_{nobs} y generar características sensibles dadas las características de identificación en P_{obs}. Se pueden usar estimadores para calcular la media y la varianza de características sensibles usando los conjuntos de datos publicados.

En este trabajo, se analizan diferentes modelos de ML como sintetizadores de datos para evaluar comparativamente la efectividad de diferentes modelos: árbol de decisión, bosque aleatorio y red neuronal. Para ello, se entrena una red neuronal con dos capas ocultas. El problema se enmarca como una clasificación de clase K, donde K es el número de valores únicos en cada una de las características sensibles. Hay K nodos en la capa de salida de la red neuronal con valor en la k-ésima neurona que representa la probabilidad de la clase k. Para generar un valor a partir de una característica particular, se muestrea un valor de clase utilizando los valores de neuronas de la capa de salida como una distribución multinomial.

La **utilidad** del conjunto de datos generado se evalúa en dos niveles diferentes. En primer lugar, se necesita evaluar las diferencias entre la distribución de los valores de los atributos

originales y la distribución de los valores de los atributos generados. En segundo lugar, es necesario evaluar la diferencia entre la calidad de la estimación de un determinado estimador para datos sintéticos y datos originales.

Sea $y \in Y$ cualquier característica sensible que se genera sintéticamente a partir de los datos originales. En el primer nivel, se calcula la divergencia KL normalizada entre la distribución de valores de y en la población original y la distribución de valores de y generados sintéticamente para estudiar la similitud entre ambas. Para m conjuntos de datos sintéticos, se considera la media de la divergencia KL normalizada con conjuntos de datos individuales. Cuanto más cerca esté el valor de 1, más similares serán los valores generados sintéticamente a los valores originales.

En el segundo nivel, un mecanismo basado en la superposición entre intervalos de confianza evalúa la efectividad de estimaciones específicas. Se estiman la media y la varianza de y usando los estimadores puntuales y se construye un intervalo de confianza del 95 % alrededor del estimador. Sea (L_s, U_s) el intervalo de confianza de los datos generados sintéticamente y (L_o, U_o) el intervalo de los datos originales. Se calcula la intersección de estos intervalos y se denota como (L_i, U_i). La medida de utilidad de superposición se calcula como se indica en la Ecuación 3.2. El valor de u es cercano a uno si se preserva la utilidad y $u = 0$ se refiere a la similitud nula de los intervalos de confianza.

$$u = \frac{(U_i - L_i)}{2 \cdot (U_o - L_o)} + \frac{(U_i - L_i)}{2 \cdot (U_s - L_s)} \tag{3.2}$$

Para estimar el **riesgo de divulgación** en el conjunto de datos generado sintéticamente, se considera que un intruso posee un vector de información t. Se supone que, además de las características sensibles, el intruso tiene información completa sobre una característica de identificación. Así pues, intenta hacer coincidir cada registro en el objetivo en t con el registro en los conjuntos de datos publicados. Para un registro $j \in t$, el intruso puede encontrar múltiples registros con una coincidencia exacta en las características de interés. Se dice que un intruso tiene éxito en la identificación del registro si encuentra solo un registro con coincidencia exacta. Para un registro $j \in t$, un intruso puede encontrar múltiples registros con el mismo valor de máxima probabilidad. Si R denota el conjunto de registros en t para el cual solo un registro en el conjunto de datos coincide con la probabilidad más alta, el conjunto R se puede descomponer en dos conjuntos mutuamente exhaustivos, T y F, que denotan un conjunto de registros con coincidencias verdaderas y falsas, respectivamente. Para evaluar el riesgo de divulgación, se calcula la tasa de coincidencia verdadera y la tasa de coincidencia falsa como se indica en la Ecuación 3.3 y en la Ecuación 3.4, respectivamente.

$$\text{Tasa de coincidencia verdadera} = \frac{|T|}{|t|} \tag{3.3}$$

$$\text{Tasa de coincidencia falsa} = \frac{|F|}{|R|} \tag{3.4}$$

Cuanto menor sea la tasa de coincidencia verdadera, mejor será el rendimiento del sintetizador de datos y lo contrario para la tasa de coincidencia falsa.

Descripción individual de los trabajos relacionados

En este apartado, se toma una evaluación experimental real. Se dispone de un conjunto de datos del censo de EE.UU. en 2003 (proporcionada por *IPUMS International*) con las caracte-

rísticas listadas en la Tabla 3.3.

Nombre del atributo	Tipo de variable
Tipo de casa	Categórica
Tamaño de la familia	Ordinal
Sexo	Categórica
Edad	Ordinal
Estado civil	Categórica
Raza	Categórica
Educación	Categórica
Empleo	Categórica
Ingresos	Ordinal
Lugar de nacimiento	Categórica

Tabla 3.3: Variables del conjunto de datos del censo de EE.UU. en 2003

El proceso comienza con la extracción de una muestra del 1 % de la población que se trata como el conjunto de datos de muestra original. Se generan sintéticamente valores para dos atributos: ingresos y edad, en el mismo orden. En este caso, 5 conjuntos de datos sintéticos para cada conjunto de datos de muestra original. Se repite este procedimiento para 500 muestras originales y se calcula la media de varias métricas en 500 iteraciones.

Para generar conjuntos de datos sintéticos, se necesita definir los límites en los valores de características que determinan si el registro contiene información confidencial. Se puede considerar, por ejemplo, que los registros que tienen un valor de ingresos superior a 70000$ y un valor de edad inferior a 26 son los que contienen información confidencial. Entonces, se generan sintéticamente valores de edad e ingresos para los registros que se ajustan a estos criterios.

Los resultados de la evaluación de la **utilidad** se presentan en la Tabla 3.4. Se observa que, en general, los conjuntos de datos sintéticos muestran un alto grado de superposición con el conjunto de datos original. En el caso de la regresión lineal, se observa una gran desviación de la media en el caso de la edad. Esto es debido a que aprende parámetros minimizando la pérdida al cuadrado en los datos de entrenamiento. El árbol de decisión y otros modelos no son propensos a sobreajustar los datos de entrenamiento debido a su orden de síntesis variable.

Característica	Sintetizador de datos	Muestra original	Datos parcialmente sintéticos		
		Media	Media sintética	Superposición	Divergencia KL normalizada
Ingresos	Regresión lineal	27112.61	27117.99	0.98	0.54
	Árbol de decisión	27143.93	27131.14	0.94	0.53
	Bosque aleatorio	27107.04	27254.38	0.95	0.58
	Red neuronal	27069.95	27370.99	0.81	0.54
Edad	Regresión lineal	49.84	24.69	0.50	0.55
	Árbol de decisión	49.83	49.83	0.90	0.56
	Bosque aleatorio	49.82	49.74	0.95	0.56
	Red neuronal	49.87	49.78	0.90	0.56

Tabla 3.4: Evaluación de la utilidad

Como ya se ha comentado, evaluar el **riesgo de divulgación** requiere un escenario, que se selecciona realizando un análisis exploratorio sobre la población. Los registros que ocurren escasamente en la población, por ejemplo, los registros de personas nacidas en el Medio Oriente con un cierto umbral de ingresos, ocurren igualmente de forma escasa en una muestra pequeña. Para evaluar estadísticamente el riesgo de divulgación, se necesita tener al menos un par de objetivos para la evaluación. Teniendo en cuenta estos requisitos, se va a suponer que un intruso está interesado en personas nacidas en Estados Unidos y con ingresos superiores a 250000$. Todas estas personas son los objetivos del intruso. Este intenta hacer coincidir cada objetivo individual con los registros en los conjuntos de datos publicados. Se considera que dos registros coinciden perfectamente si las personas que representan los registros nacieron en los EE.UU., tienen ingresos superiores a 250000$ y la edad de la persona en el conjunto de datos está dentro de la tolerancia de 2 en comparación con la persona objetivo. La evaluación de riesgos se presenta en la Tabla 3.5.

Sintetizador de datos	Tasa de coincidencia verdadera	Tasa de coincidencia falsa
Regresión lineal	0,06	0,82
Árbol de decisión	0,18	0,68
Bosque aleatorio	0,35	0,50
Red neuronal	0,03	0,92

Tabla 3.5: Evaluación del riesgo de divulgación

Se observa que las redes neuronales son mejores para reducir el riesgo de divulgación que el resto de los sintetizadores de datos. No muy por detrás de las redes neuronales se encuentran los resultados de la regresión lineal, que ganan con diferencia a los sintetizadores restantes.

Por último, se analiza comparativamente la eficiencia de la generación utilizando los diferentes sintetizadores de datos. El **tiempo de ejecución**, en segundos, para generar 5 conjuntos de datos sintéticos se informa en la Tabla 3.6.

Sintetizador de datos	Regresión lineal	Árbol de decisión	Bosque aleatorio	Red neuronal
Tiempo	0,040	0,048	3,350	0,510

Tabla 3.6: Evaluación del tiempo de ejecución (en segundos)

Es conveniente mencionar que todos los programas se ejecutan en una máquina Linux con cuatro núcleos de 2,40 GHz y procesador Intel® Core i7™ con 8 GB de memoria. La máquina está equipada con dos GPU Nvidia GTX 1080. Además, como lenguaje de secuencias de comandos se utiliza Python® 2.7.6. Sabiendo esto, se observa que las redes neuronales logran el riesgo de divulgación bajo a costa de un valor alto del tiempo de ejecución.

Toda esta información individual se recoge en la Tabla 3.7 en forma de resumen de los métodos comparados y su rendimiento con las métricas utilizadas para evaluarlos.

Métrica \ Sintetizador de datos		Regresión lineal	Árbol de decisión	Bosque aleatorio	Red neuronal
Utilidad retenida	Superposición	0.74	0.92	0.95	0.855
	Divergencia KL normalizada	0.545	0.545	0.57	0.55
Riesgo de divulgación	Tasa de coincidencia V	0.06	0.18	0.35	0.03
	Tasa de coincidencia F	0.82	0.68	0.50	0.92
Tiempo de generación		0.040	0.048	3.350	0.510

Tabla 3.7: Resumen comparación cuantitativa

Discusión

En este estudio comparativo de técnicas de generación de conjuntos de datos sintéticos utilizando diferentes sintetizadores de datos, a saber, regresión lineal, árbol de decisión, bosque aleatorio y red neuronal, se evalúa la utilidad utilizando estimadores estadísticos y se aborda el aspecto de la privacidad calculando el riesgo de divulgación. Además, se compara el tiempo de ejecución de cada uno de los modelos.

El análisis muestra que las redes neuronales son competitivamente efectivas en comparación con otros métodos en términos de utilidad y privacidad, pero logran esta eficacia a costa del tiempo de ejecución. Siguiendo a éstas, se encontrarían el resto de sintetizadores a la par, ya que la regresión lineal obtiene mejores resultados que el árbol de decisión y el bosque aleatorio en cuanto a riesgo de divulgación se refiere pero peores en términos de utilidad retenida. Los resultados en utilidad retenida y riesgo de divulgación de la regresión lineal y el árbol de decisión no logran alcanzar a los de las redes neuronales, pero lo consiguen en menos tiempo de ejecución. También se considera preciso mencionar que los números relativos al bosque aleatorio son decepcionantes debido a que se trata de un modelo cuyo objetivo es mejorar el rendimiento del árbol de decisión, al cual solo supera, y por un par de centésimas, en términos de utilidad retenida. Este sorprendente derrota puede deberse sin más a que el ejemplo no favorece a dicho modelo.

También se realizan experimentos para generar conjuntos de datos completamente sintéticos. El conjunto de datos completamente sintético se genera mediante el uso de P_{nobs} como un conjunto de datos reservado y la generación sintética de valores de características sensibles para todos los registros en P_{obs}. Por lo tanto, los conjuntos de datos totalmente sintéticos brindan una mayor efectividad para reducir el riesgo de divulgación. Dado que se necesita generar sintéticamente P_{obs} de toda la población, los conjuntos de datos totalmente sintéticos requieren un mayor tiempo de cálculo y muestran una menor retención de la utilidad, es decir, son menos útiles.

3.3.3. Decisión

En el caso de este proyecto, se ha elegido generar datos tabulares totalmente sintéticos mediante una técnica híbrida, es decir, un algoritmo que combina dos de las técnicas estudiadas en la sección 3.2; en particular, las dos primeras formas, explicadas en las secciones 3.2.1 y 3.2.2. La elección de las técnicas (proceso estocástico y generación basada en reglas) procede de la comparación cualitativa, a partir de la cual se descartan los modelos de ML por su complejidad computacional. La opción de generar datos totalmente sintéticos procede de la comparación cuantitativa, que concluye que estos son más efectivos que los datos parcialmente sintéticos en cuanto a la reducción del riesgo de divulgación. Esto permite obtener un generador completo, ya que el contenido de la información es justamente la métrica en la que las dos técnicas elegidas flaquean en la comparación cualitativa mostrada.

3.4. Otros recursos de interés

En esta sección, se analizan las herramientas y recursos existentes en el panorama científico-técnico ya consolidado que permiten materializar la visión del proyecto. Se comienza haciendo una revisión general de las técnicas numéricas y estadísticas utilizadas en la generación de datos sintéticos (sección 3.4.1) para continuar con la caracterización de las librerías existentes en Python, lenguaje elegido para la implementación del sistema *software* asociado a este proyecto, disponibles para este fin (sección 3.4.2).

3.4.1. Técnicas numéricas para la generación de datos sintéticos tabulares

Como se explicó anteriormente, existen diferentes tipos de datos sintéticos: estructurados y no estructurados. Esta subsección se centra en el campo de especialización de este proyecto,

la generación de datos sintéticos tabulares (estructurados), aunque las técnicas estadísticas que se mencionan a continuación (distribuciones marginal y conjunta, tomadas de [67]) también se estudian y utilizan para la generación de datos sintéticos no estructurados.

El objetivo final de la generación de datos tabulares sintéticos es tomar una fuente de datos original y crear una sintética con propiedades estadísticas similares. Tener propiedades estadísticas similares significa que es necesario reproducir la distribución en la medida en que, en última instancia, se debería poder inferir la misma conclusión de ambas versiones de los datos. También se necesita mantener la estructura de los datos originales.

Para hacerlo, hay que aprender una distribución o proceso aproximado compatible con los datos originales (es decir, un modelo generativo) que luego se pueda usar para muestrear datos sintéticos estructural y estadísticamente comparables. Este proceso de aprendizaje y muestreo puede observarse en la Figura 3.9 explicado sobre un conjunto de datos tabular. En esta tabla, se representa el listado de empleados de una empresa, donde cada uno posee un identificador, usa un medio de transporte y pertenece a un departamento.

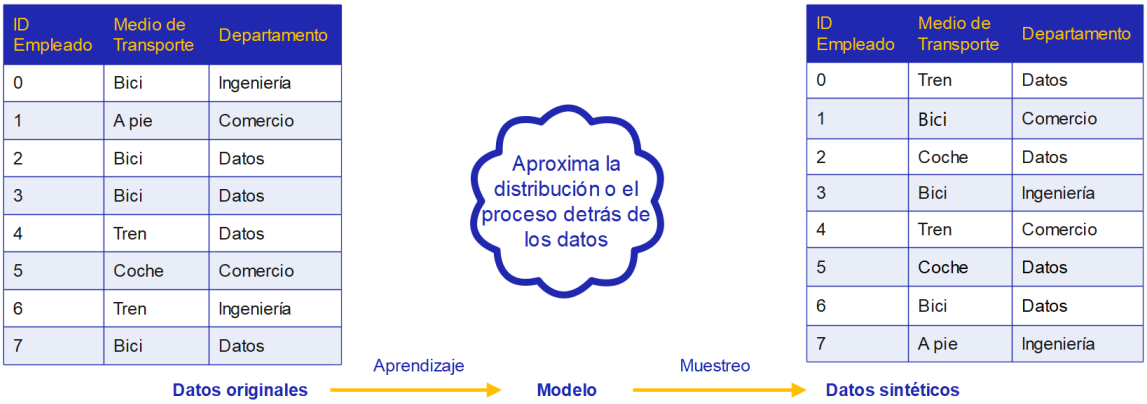


Figura 3.9: Esquema de la generación de datos sintéticos

Un enfoque simple para aprender las probabilidades conjuntas sería contar la ocurrencia de cada valor en cada columna de forma independiente. Los resultados son distribuciones discretas que se convierten en el modelo. En la Figura 3.10 se pueden observar los sectores circulares representando la distribución de cada uno de los atributos.

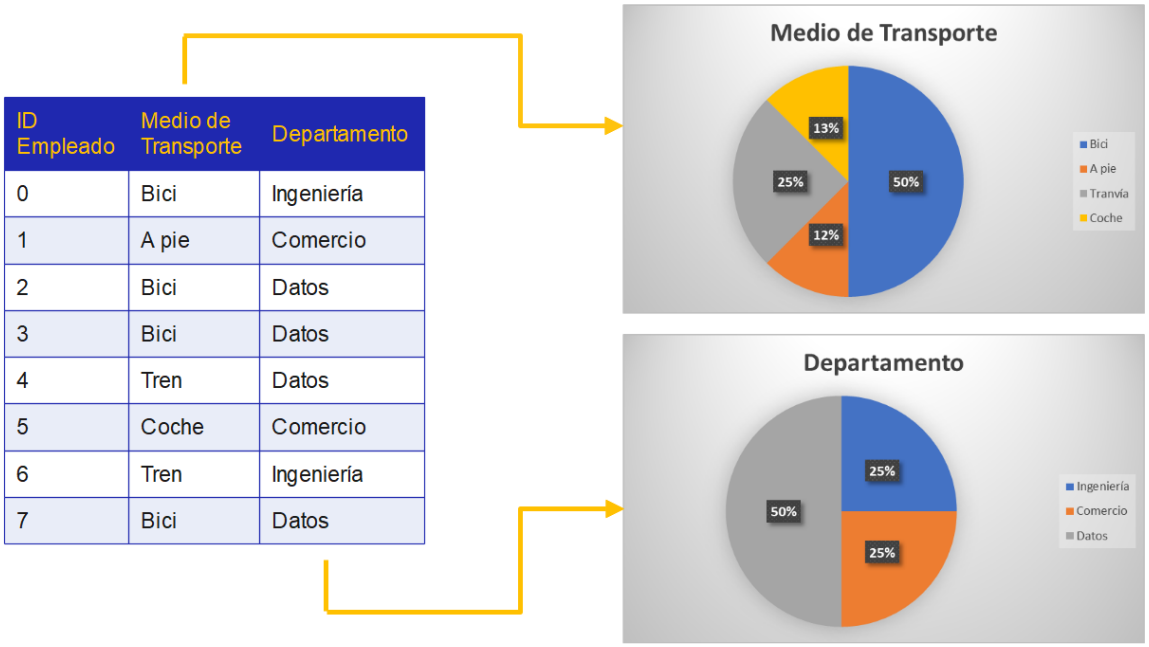


Figura 3.10: Ejemplo de las distribuciones marginales

Sin embargo, este enfoque perdería posibles conexiones entre las columnas. Para incluir estos patrones, una solución podría ser contar las combinaciones que ocurren. Como resultado, en lugar de varias distribuciones marginales, se obtiene una distribución conjunta que se puede utilizar para crear la tabla sintética. La distribución conjunta del ejemplo aparece representada en la Figura 3.11.

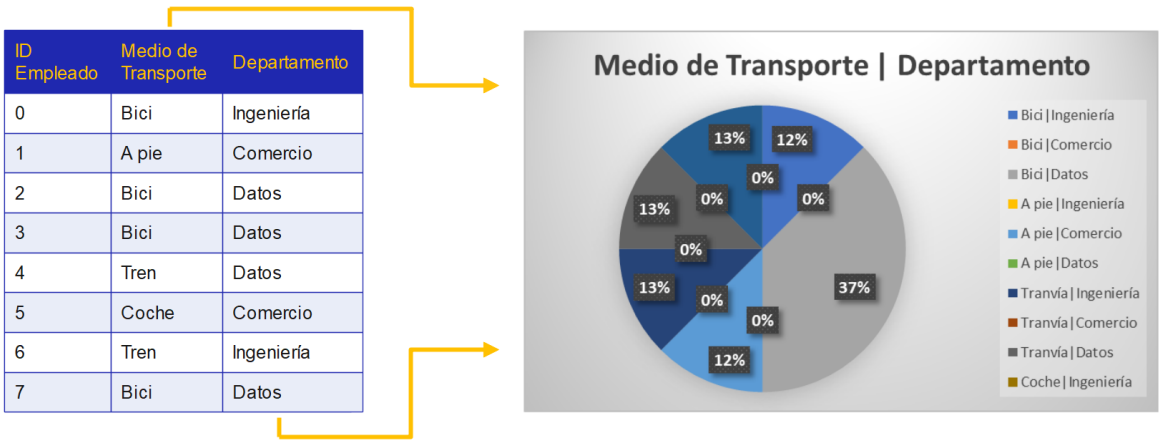


Figura 3.11: Ejemplo de la distribución conjunta

Teóricamente es un enfoque válido, pero no escalaría si aumentamos la complejidad del conjunto de datos; cuantas más columnas se agregan, más combinaciones aparecen. Por eso, se necesita un modelo más robusto para abordar la complejidad de los datos.

Se puede usar este enfoque tan simple cuando los datos son básicos y no contienen dependencias demasiado complicadas. Con una tabla simple y muy pocas columnas, y ninguna o pocas dependencias, un modelo muy simplista puede ser una forma rápida y fácil de obtener datos sintéticos. A medida que los datos crecen en complejidad, se debe actualizar el tipo de modelo utilizado. Las redes neuronales están bien adaptadas para simplificar los problemas de transformación porque son buenas para encontrar patrones en los datos. Sus funciones de transformación generan una distribución más fácil de aprender sin sacrificar la información.

3.4.2. Librerías en Python

Python es un lenguaje de programación interpretado, interactivo y orientado a objetos que incorpora módulos, excepciones, tipificación dinámica, tipos de datos dinámicos de muy alto nivel y clases. Admite múltiples paradigmas de programación más allá de la programación orientada a objetos, como la programación procedimental y funcional. Tiene interfaces para muchas llamadas al sistema y bibliotecas, así como para varios sistemas de ventanas, y es extensible en C o C++. Cuenta con una sintaxis muy clara y es utilizable como lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Finalmente, Python es portable: se ejecuta en muchas variantes de *Unix*, incluyendo *Linux* y *macOS*, y en *Windows* [68]. Por todo esto, es uno de los lenguajes de programación más populares en la actualidad, especialmente para la ciencia de datos. Por ello, el producto de este proyecto se desarrolla en Python.

Para modelado, análisis y visualización estadísticos A continuación, se describen algunos de los marcos y métodos existentes para hacer modelado, análisis y visualización estadísticos dentro de una plataforma Python y que, por tanto, son recursos potenciales para este proyecto.

Numpy [66] es el estándar de facto para la computación numérica en Python, utilizado como base para construir bibliotecas más avanzadas para aplicaciones de ciencia de datos y ML como TensorFlow o Scikit-learn. Aunque la mayoría de las discusiones relacionadas con Numpy se centran en sus rutinas de álgebra lineal, ofrece un conjunto decente de funciones de modelado estadístico para realizar estadísticas descriptivas básicas y generar variables aleatorias basadas en varias distribuciones discretas y continuas.

Los científicos de datos deben poder visualizar rápidamente varios tipos de datos para hacer observaciones, detectar valores atípicos, recopilar información, obtener patrones de investigación y, lo que es más importante, comunicar los resultados para la toma de decisiones comerciales. Existen dos poderosas bibliotecas de Python para esta tarea de visualización. Por un lado, **Matplotlib** [66] es la biblioteca base más utilizada en Python para la visualización general. Por otro lado, **Seaborn** [66] es otra biblioteca de Python que se construye sobre Matplotlib, proporcionando API directas para visualizaciones estadísticas dedicadas; es, por ello, una de las favoritas entre los científicos de datos. Algunas de las gráficas de modelado estadístico avanzado que Seaborn proporciona son mapas de calor, violines, diagramas de dispersión con regresión lineal, ajuste e intervalos de confianza, gráficos de pares y gráficos de correlación que muestran la dependencia mutua entre todas las variables de una tabla de datos (con varias filas y columnas) y gráficos con facetas (es decir, visualizar una relación entre dos variables que dependen de más de una variable).

SciPy [66] es un ecosistema de *software* de código abierto para matemáticas, ciencias e ingeniería. De hecho, Numpy y Matplotlib son componentes de este ecosistema. Específicamente en el modelado estadístico, SciPy posee una gran colección de métodos y clases rápidos, potentes

y flexibles. Destaca su uso para estadísticas inferenciales. Con SciPy se puede generar variables aleatorias a partir de una amplia selección de distribuciones estadísticas discretas y continuas (Binomial, Normal, Beta, Gamma, t de Student, etc.), calcular la frecuencia y resumir las estadísticas de los conjuntos de datos multidimensionales, ejecutar pruebas estadísticas populares (como t-test, chi-cuadrado, Kolmogorov-Smirnov, prueba de rango de Mann-Whitney, wilcoxon rank-sum, etc.), realizar cálculos de correlación (como el coeficiente de Pearson, ANOVA, estimación de Theil-Sen, etc.) y calcular medidas estadísticas de distancia (como la distancia de Wasserstein y la distancia de energía).

Más allá de la computación de estadísticas descriptivas e inferenciales básicas, entramos en el ámbito del modelado avanzado, por ejemplo, regresión multivariante, modelos aditivos generalizados, pruebas no paramétricas, análisis de supervivencia y durabilidad, modelado de series temporales, imputación de datos con ecuaciones encadenadas, etc. El paquete **Statsmodels** [66] permite realizar todos estos análisis. Los modelos de estadística permiten la sintaxis de fórmula de estilo R para muchas API de modelado y también producen tablas detalladas con valores importantes para el modelado estadístico, como valores p, R cuadrado ajustado, etc.

Para generación de datos sintéticos Por otro lado, hay tres bibliotecas que los científicos de datos pueden utilizar para generar datos sintéticos, Scikit-learn, SymPy y Pydbgen.

Scikit-learn [66] es una de las bibliotecas de Python más utilizadas para tareas de ML y la más utilizada para el ML clásico. Se podría incluir también en la discusión de la modelización estadística debido a que muchos algoritmos clásicos de ML se pueden clasificar como técnicas de aprendizaje estadístico. Scikit-learn presenta varios algoritmos de clasificación, regresión y agrupación en clústeres, incluidas máquinas vectoriales de soporte (SVM), bosques aleatorios, k-medias y DBSCAN. Está diseñado para interoperar sin problemas con las bibliotecas numéricas y científicas Numpy y SciPy, proporcionando una gama de algoritmos de aprendizaje supervisados y no supervisados a través de una interfaz consistente. La biblioteca Scikit-learn también es lo suficientemente robusta para su uso en sistemas de producción debido a su comunidad de soporte. Con Scikit-learn se pueden realizar tareas avanzadas de aprendizaje estadístico como analizar modelos estadísticos en una cadena, generar datos aleatorios de regresión y clasificación para probar algoritmos, realizar varios tipos de codificación / transformación en los datos de entrada, buscar hiperparámetros para algoritmos complejos como SVM, etc.

Con **SymPy** [66], los usuarios pueden especificar las expresiones simbólicas para los datos que desean crear, lo que ayuda a los usuarios a crear datos sintéticos de acuerdo con sus necesidades. Los datos categóricos también se pueden generar utilizando la biblioteca **Pydbgen** [66]. Los usuarios pueden generar nombres aleatorios, números de teléfono internacionales, direcciones de correo electrónico, etc. fácilmente utilizando esta biblioteca [66].

En la Figura 3.12 se presenta de forma visual cada librería y lo que esta aporta, así como las relaciones entre ellas.

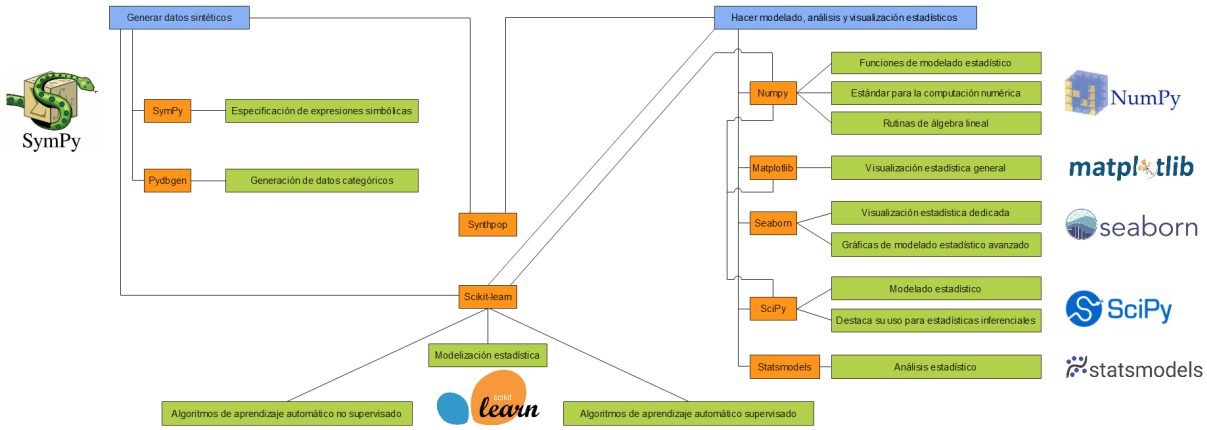


Figura 3.12: Librerías de Python

Capítulo 4

Descripción y desarrollo de la propuesta

Este capítulo describe los resultados principales del proceso de ingeniería de *software* que se sigue para construir el producto, esto es, los procesos de análisis (sección 4.2) y diseño (sección 4.3) del producto. Antes de tratar ambos procesos, se explica, en la sección 4.1, el modelo matemático que representa la pandemia y que se usa como regla para la posterior generación de datos sintéticos. Por último en este capítulo, se dedica un esfuerzo a explicar la implementación del sistema (sección 4.4) y a realizar manuales de ayuda al usuario (sección 4.5).

En este proyecto se desarrolla un sistema *software* que genera datos tabulares totalmente sintéticos acerca de la pandemia COVID-19 para series temporales que se pueden caracterizar según unos parámetros.

4.1. Modelo SEIR

El sistema *software* desarrollado en este proyecto está basado en una modelización matemática de la epidemia de coronavirus o COVID-19, el modelo epidemiológico SEIR (una adaptación del modelo SIR, que fue propuesto por W. O. Kermack y A. G. McKendrick en 1927). Lo que se explica a continuación proviene del estudio de la fuente [69].

Este modelo plantea que en una población de tamaño fijo N en la que se ha desatado una epidemia que se propaga mediante contagio, en un tiempo t los individuos pueden estar en cuatro estados distintos:

- Susceptibles, $S(t)$, que recoge a los individuos susceptibles de ser infectados.
- Expuestos, $E(t)$, que recoge a los individuos que portan la enfermedad pero que, al hallarse en su periodo de incubación, no muestran síntomas y aún no pueden infectar a otros.
- Infectados, $I(t)$, que recoge a los individuos ya infectados.
- Recuperados, $R(t)$, que recoge a los individuos ya recuperados del virus.

Para ser precisos con lo que se denota, conviene aclarar que, si un individuo no presenta síntomas, pero sí puede contagiar a otros se contabiliza en $I(t)$, no en $E(t)$. También cabe mencionar que, una vez que se recuperan, los individuos son inmunes y ya no vuelven a ser susceptibles (de hecho, también se puede pensar que no todos los individuos se recuperan, sino que pueden morir a causa de la enfermedad: ambos tipos de casos están recogidos en $R(t)$, y ya

no afectan al desarrollo de la epidemia). Es decir, para simplificar el estudio, se considera que un individuo solo puede tener el virus una vez en su vida, a pesar de que es bien conocido que en la realidad esto no es así debido, principalmente, a las diferentes variantes que ha presentado el virus. Por eso, algunos investigadores se refieren a los individuos en R como "removidos", para evitar el contrasentido de llamar "recuperados" a los fallecidos [70].

En el modelo SEIR existen tres parámetros:

- β, llamado tasa de transmisión, de manera que $\frac{1}{\beta}$ mide la probabilidad de que un susceptible se infecte cuando entra en contacto con un infectado;
- γ, llamado tasa de recuperación, de manera que el periodo medio de recuperación es $\frac{1}{\gamma}$;
- σ, de forma que $\frac{1}{\sigma}$ es el tiempo promedio de incubación.

Los dos primeros parámetros definen el parámetro $R_0 = \frac{\beta}{\gamma}$ que se llama tasa básica de reproducción y representa el número de nuevos infectados producidos por un infectado si toda la población es susceptible.

En el modelo SEIR, el flujo entre los distintos grupos, que deben cumplir $S(t)+E(t)+I(t)+R(t) = N$ (es decir, los estados son disjuntos y la población total N no varía, no se producen nacimientos y las defunciones por coronavirus se almacenan en R), se representa mediante la Figura 4.1.

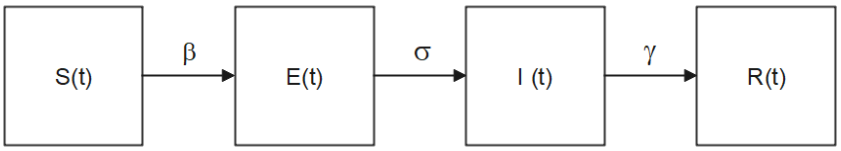


Figura 4.1: "Diagrama de flujo" del modelo SEIR

Teniendo todo esto en cuenta, las ecuaciones diferenciales ordinarias que forman el sistema del modelo SEIR básico son las siguientes:

$$S'(t) = -\beta \cdot \frac{S(t) \cdot I(t)}{N} \tag{4.1}$$

$$E'(t) = \beta \cdot \frac{S(t) \cdot I(t)}{N} - \sigma \cdot E(t) \tag{4.2}$$

$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) \tag{4.3}$$

$$R'(t) = \gamma \cdot I(t) \tag{4.4}$$

El modelo matemático utilizado representa el comportamiento de un agente infeccioso en una población teniendo en cuenta solo algunos factores (probabilidad de infección, sujetos susceptibles...), mientras que este comportamiento en la realidad es mucho más complejo y sujeto a una gran variedad de factores (algunos de ellos muy difíciles o virtualmente imposibles de modelar: reinfección, mortalidad, variación en la población, aplicación de medidas concretas, patrones en la interacción entre individuos...). Este hecho motiva la aparición de modelos más complejos con

mayor cantidad de parámetros, que aproximan de forma más fidedigna el comportamiento de una pandemia en la realidad, pero cuya complejidad excede el alcance de este trabajo. En el caso del coronavirus, bastantes de los estudios ya publicados [71] [72] [73] introducen funciones o ecuaciones adicionales con el objeto de tener en cuenta el comportamiento del inicio de la epidemia en Wuhan en diciembre de 2019, cuando los humanos se infectaban directamente del contacto con animales, así como los desplazamientos extraordinarios de población fruto de las festividades chinas de la época, y otras variables. En este modelo SEIR [69] se prescinde de todo eso con el fin de simplificar el lado matemático del sistema y alcanzar los objetivos del proyecto de acuerdo con las restricciones académicas en las que se desarrolla.

Siguiendo lo publicado en [71], los valores de los parámetros que mejor modelizan la evolución del coronavirus o COVID-19 son $\gamma = \frac{1}{5}$ y $\sigma = \frac{1}{7}$. Estos valores son los que tomará por defecto el sistema para estos parámetros cuando el usuario no los introduzca o lo haga erróneamente. Lo más complicado es estimar β, ya que no se sabe cuántas personas infectadas asintomáticas pueden estar infectando a otros.

Diversos estudios sobre el comportamiento de la enfermedad en China, antes de que se adoptaran medidas drásticas de aislamiento de la población (como el ya citado [71]), sugieren que $0{,}59 \leq \beta \leq 1{,}68$ (en unidades/día), lo cual daría un $2{,}95 \leq R_0 \leq 8{,}4$, en ambos casos bastante elevado (un reciente artículo en la revista Lancet [72] toma $R_0 = 2{,}68$).

Hasta ahora, en este modelo se ha considerado que la tasa de infección β es constante, pero este parámetro se puede ajustar artificialmente si se adoptan medidas de contención (protección y aislamiento), y si la población las acepta y las cumple. Así, existen diversos estudios con parámetros β que van variando en el tiempo. Por ejemplo [73], con una función decreciente respecto al tiempo

$$\beta(t) = \beta_0 \cdot ((c_0 - c_b) \cdot \exp -r_1 \cdot t + c_b) \tag{4.5}$$

donde β_0 es la tasa de infección sin medidas de contención y las constantes c_0 y c_b aluden a las tasas de contacto (fruto de los aislamientos de la población); o, también [71],

$$\beta(t) = \beta_0 \cdot (1 - \alpha(t)) \cdot \left(1 - \frac{D(t)}{N}\right)^{\kappa} \tag{4.6}$$

donde β_0 es la tasa de infección sin medidas, $\alpha(t)$ (con valores en el intervalo $[0, 1]$) es el resultado de las acciones gubernamentales, $D(t)$ es la sensación pública de riesgo como consecuencia de los casos críticos y muertes conocidos, y κ mide la intensidad de la reacción de los individuos. Típicamente, $\alpha(t)$ es una función constante a trozos (las medidas se toman en momentos concretos). Además, cabe mencionar que la idea del factor $\left(1 - \frac{D(t)}{N}\right)^{\kappa}$ con un κ elevado es que, cuando la preocupación es mucha, el factor es muy cercano a 0, la gente se aísla incluso voluntariamente y $\beta(t)$ es muy pequeño; por el contrario, el factor es cercano a 1 y tiene escasa influencia si la preocupación social es poca.

Como la realidad político-social china y la occidental no es la misma, en Occidente no se pueden usar sus mismos parámetros, pero sí sus mismas ideas de modelización. Cabe reiterar que todo esto son estimaciones; es imposible conocer el futuro, pero sí se puede analizar cómo las medidas afectan a la evolución de la pandemia; en particular, los cambios en el valor de α. Qué α va a ser el que se obtiene con unas medidas concretas tomadas por un gobierno es imposible de conocer de antemano, sólo podría ser estimado a posteriori, a la vista de lo que haya ocurrido; pero, sea cual sea, su efecto va a ser positivo, y, si en la evaluación diaria de los

datos epidemiológicos se ve que no es suficiente para lo que se pretende conseguir, se pueden endurecer las medidas.

El sistema *software* construido en este proyecto parte del modelo SEIR como forma de simular series temporales. Sin embargo, es el usuario el que introduce los parámetros en cada caso. El usuario debe escoger si desea obtener datos supuesta la no adopción de medidas gubernamentales de contención o no pero, antes de esta elección, existen 4 parámetros comunes a ambas opciones de los que también éste ha de hacerse cargo. La elección del caso lleva a la solicitud, por parte del sistema, de distintos parámetros en cada caso. Se plantea la opción de no rellenar dichos parámetros, generando datos a partir de los valores por defecto que se determinan a continuación y que han sido explicados anteriormente. En cualquiera de los dos casos, se solicita el tamaño de la población, N (que, por defecto, es 100000 personas), y el número de días o número de datos a generar menos uno, L (que, por defecto, es 120 días). Se generan $L + 1$ registros, desde el día 0 hasta el día L.

- Parámetros comunes a ambos casos:
  - Parámetros solicitados: N, L, γ y σ
  - Valores por defecto: $N = 100000$, $L = 120$, $\gamma = \frac{1}{5} = 0{,}2$, $\sigma = \frac{1}{7} = 0{,}15$
- Caso 1 (Sin medidas de contención):
  - Parámetros solicitados: Ninguno
- Caso 2 (Con medidas de contención):
  - Parámetros solicitados: β_0 y t_0
  - Valores por defecto: $\beta_0 = 1$ y $t_0 = 0$

Los valores por defecto para γ y σ han sido escogidos como consecuencia de los estudios citados anteriormente. Sin embargo, la razón por la que se han elegidos los valores para el resto de parámetros, N, L, β_0 y t_0, es que existen ejemplos realizados en el estudio con dichos valores. De esta manera ha sido posible verificar que el código del sistema funciona correctamente antes de introducir la aleatoriedad sobre el parámetro β.

En este sistema se pretende usar una técnica de generación de datos sintéticos híbrida entre los dos primeros grupos mencionados en la sección 3.2, es decir, un proceso estocástico basado en reglas. Para la consecución de este objetivo, se añade cierta estocasticidad entre las reglas prefijadas por el modelo SEIR. Por ello, el valor del parámetro β en cada uno de los casos se obtiene mediante un proceso aleatorio, sin olvidar la desigualdad $0{,}59 \leq \beta \leq 1{,}68$.

En el primer caso, el parámetro β consiste en una constante obtenida de forma totalmente aleatoria. En el segundo caso, $\beta(t)$ es una función definida a trozos; el primer trozo, con t perteneciente a $0 \leq t \leq t_0$, es igual a la constante β_0 introducida por teclado mientras que el segundo y último trozo, con t perteneciente a $t_0 + 1 \leq t \leq L + 1$, se obtiene de manera pseudoaleatoria, sumando un valor aleatorio en el intervalo $[-0{,}025, 0{,}025]$ al anterior valor de $\beta(t)$, esto es, a $\beta(t - 1)$.

Como resultado, el usuario obtiene una serie temporal, cuyos registros se ajustan al diseño planteado en la subsección 4.3.3.

4.2. Análisis de *software*

En esta sección se describe, en primer lugar, al usuario del sistema (subsección 4.2.1) para después continuar con los requisitos de usuario (subsección 4.2.2), funcionales y no funcionales (sección 4.2.3) característicos del sistema.

4.2.1. Actores

Solo habrá un tipo de usuario con acceso a toda la aplicación sin necesidad de registro, como se describe en la Tabla 4.1.

ID	Actor	Descripción
A1	Usuario General	Cualquier usuario que accede al sistema y, por tanto, está interesado en generar datos sintéticos sobre la pandemia.

Tabla 4.1: Actores

4.2.2. Requisitos de usuario

Se trata de las acciones que puede realizar el usuario.

Diagrama de casos de uso Estas acciones se indican en el siguiente diagrama de casos de uso (Figura 4.2).

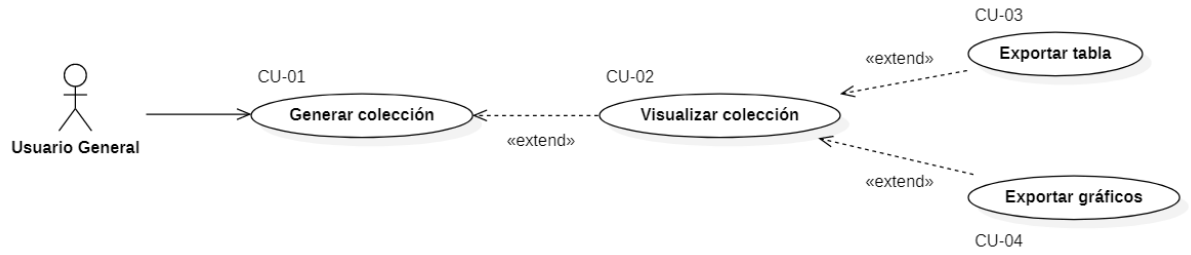


Figura 4.2: Diagrama de casos de uso

Especificación de casos de uso En la Tabla 4.2 se explica el primer caso de uso que se ha identificado en la Figura 4.2.

CU-01	Generar colección
Descripción	El Usuario General introduce los 4 parámetros generales, esto es, N, L, γ y σ, antes de seleccionar uno de los dos casos posibles (Caso 1: sin medidas gubernamentales de contención o Caso 2: con medidas gubernamentales de contención) y, dependiendo del caso que elija, introduce los parámetros que permiten caracterizar la serie temporal a generar sintéticamente, esto es, β_0 y t_0 en el segundo caso.

Autor	Analista de datos		
Actor	Usuario General		
Secuencia Normal	Caso 1	Paso 1	El usuario proporciona los parámetros N, L, γ y σ.
		Paso 2	El sistema pone dos casos a disposición del usuario: Caso 1, sin medidas de contención; Caso 2, con medidas de contención.
		Paso 3	El usuario selecciona el Caso 1.
		Paso 4	El sistema muestra un resumen editable de los parámetros introducidos y el caso escogido.
		Paso 5	El usuario solicita generar la colección de datos sintéticos.
		Paso 6	El sistema genera la colección de datos de acuerdo con la información proporcionada y habilita su visualización.
		Paso 7	Si el usuario selecciona dicha visualización, se realiza el caso de uso CU-02: Visualizar colección.
		Paso 8	El caso de uso finaliza satisfactoriamente.
Flujos Alternativos	FA1	Paso 3'	El usuario selecciona el Caso 2 y se continúa con el Paso 4'.
		Paso 4'	El sistema muestra un resumen editable de los parámetros introducidos y el caso escogido, solicita los parámetros β_0, y t_0 necesarios para generar la colección y se continúa con el Paso 5'.
		Paso 5'	El usuario proporciona los parámetros β_0, y t_0, solicita generar la colección de datos sintéticos y se continúa con el Paso 6.
	FA2	Paso 5"	El usuario edita los valores introducidos anteriormente, solicita generar la colección de datos sintéticos y se continúa con el Paso 6.
Excepciones	E1	Paso 1'	El usuario no introduce un dato (introduce un valor nulo) o introduce un dato no válido (fuera de las restricciones impuestas), selecciona un caso y se continúa con el Paso 2'.
		Paso 2'	El sistema muestra el mensaje "Por invalidez o ausencia de respuesta, se procede a usar el valor por defecto", rellena los valores con los seleccionados por defecto, muestra un resumen editable de los parámetros introducidos y el caso escogido y se continúa con el Paso 5.

	E2	Paso 5” Paso 5”’	El usuario no introduce un dato (introduce un valor nulo) o introduce un dato no válido (fuera de las restricciones impuestas), solicita generar la colección de datos sintéticos y se continúa con el Paso 6’.
		Paso 6’	El sistema muestra el mensaje “Por invalidez o ausencia de respuesta, se procede a usar el valor por defecto”, rellena los valores con los seleccionados por defecto, genera la colección de datos de acuerdo con la información proporcionada, habilita su visualización y se continúa con el Paso 7.

Tabla 4.2: Especificación CU-01

En la Tabla 4.3 se explica el segundo caso de uso que se ha identificado en la Figura 4.2, el cual extiende al primero.

CU-02	Visualizar colección	
Descripción	El usuario visualiza la colección generada y la comparación de los datos generados respecto a una serie real, es decir, puede visualizar además una evaluación de la colección, obtenida tras comparar ésta respecto a series reales de España y Reino Unido	
Autor	Analista de datos	
Actor	Usuario General	
Secuencia	Paso 1	El usuario solicita visualizar la colección generada.
	Paso 2	El sistema muestra la colección de datos sintéticos y habilita la visualización de la evaluación con datos reales de España y Reino Unido.
	Paso 3	El usuario solicita visualizar la evaluación con datos reales.
	Paso 4	El sistema muestra la evaluación con datos reales.
	Paso 5	El caso de uso finaliza satisfactoriamente.

Tabla 4.3: Especificación CU-02

En la Tabla 4.4 se explica el tercer caso de uso que se ha identificado en la Figura 4.2, el cual extiende al segundo.

CU-03	Exportar tabla		
Descripción	El usuario exporta la colección generada en forma tabular con formato Valores Separados por Comas o *Comma Separated Values* (CSV)		
Autor	Analista de datos		
Actor	Usuario General		
Secuencia Normal	Paso 1	El usuario proporciona el nombre del archivo (sin la extensión) donde se va a almacenar la tabla y solicita exportarla.	
	Paso 2	El sistema descarga la tabla en formato CSV con el nombre proporcionado y guarda el archivo en la carpeta del proyecto.	
	Paso 3	El caso de uso finaliza satisfactoriamente.	
Excepciones	E1	Paso 1'	El usuario no introduce el nombre del archivo pero solicita descargar la tabla y se continúa con Paso 2'.
		Paso 2'	El sistema muestra el mensaje "Por invalidez o ausencia de respuesta, se procede a usar el nombre por defecto", descarga la tabla en formato CSV con el nombre `tabla.csv`, guarda el archivo en la carpeta del proyecto y se continúa con Paso 3.

Tabla 4.4: Especificación CU-03

En la Tabla 4.5 se explica el cuarto caso de uso que se ha identificado en la Figura 4.2, el cual extiende al segundo.

CU-04	Exportar gráficos	
Descripción	El usuario exporta alguna de las representaciones gráficas generadas con formato PNG	
Autor	Analista de datos	
Actor	Usuario General	
Secuencia Normal	Paso 1	El usuario solicita descargar la gráfica.
	Paso 2	El sistema descarga la gráfica como una imagen en formato PNG con el nombre `newplot.png` y guarda el archivo en la carpeta de descargas
	Paso 3	El caso de uso finaliza satisfactoriamente

Tabla 4.5: Especificación CU-04

4.2.3. Requisitos funcionales y no funcionales

En esta subsección, se enumeran los requisitos funcionales y no funcionales del sistema en cuestión.

Requisitos funcionales Se trata de las funciones que el sistema debe proporcionar para poder satisfacer las necesidades del usuario y poder llevar a cabo los casos de uso. Aparecen descritas en la Tabla 4.6.

ID	Descripción
RF-01	El sistema mostrará información de ayuda al usuario, información acerca de las opciones que se le ofertan, lo que significan los parámetros a introducir en cada caso y un "manual de uso" de la aplicación.
RF-02	El sistema solicitará al usuario los parámetros N, L, γ y σ.
RF-03	El sistema almacenará los parámetros N, L, γ y σ introducidos por el usuario.
RF-04	El sistema comprobará si los parámetros N, L, γ y σ introducidos por el usuario son válidos (son del tipo adecuado y respetan las restricciones explicadas en la sección 4.1) y, en caso de no serlo, cambiará su valor al establecido por defecto y mostrará el mensaje "Por invalidez o ausencia de respuesta, se procede a usar el valor por defecto".
RF-05	El sistema solicitará al usuario el caso.
RF-06	El sistema almacenará el caso introducido por el usuario
RF-07	El sistema mostrará un resumen editable de los parámetros introducidos y del caso elegido por el usuario.
RF-08	El sistema solicitará al usuario los parámetros β_0 y t_0 (Caso 2).
RF-09	El sistema almacenará los parámetros introducidos (nuevos o cambiados) y el caso elegido por el usuario.
RF-10	El sistema comprobará si los parámetros N, L, γ y σ (Caso 1) o $N, L, \gamma, \sigma, \beta_0$ y t_0 (Caso 2) introducidos por el usuario son válidos (son del tipo adecuado y respetan las restricciones explicadas en la sección 4.1) y, en caso de no serlo, cambiará su valor por el establecido por defecto y mostrará el mensaje "Por invalidez o ausencia de respuesta, se procede a usar el valor por defecto".
RF-11	El sistema ofrecerá la opción de editar lo introducido hasta el momento por el usuario en cada paso.
RF-12	El sistema generará la colección de datos de acuerdo con la información proporcionada y habilitará la visualización de la colección de datos sintéticos generados
RF-13	El sistema mostrará, en forma tabular y gráfica, los datos secuenciales sintéticos calculados a partir de los parámetros de entrada y habilitará la evaluación con datos reales de España y Reino Unido.

RF-14	El sistema mostrará el valor de u 3.2 para la comparación con los datos reales y 4 gráficos: ▪ Un gráfico con la representación de las curvas $E(t) + I(t) + R(t)$ (datos sintéticos) por un lado y los casos totales acumulados reales de COVID-19 por el otro. ▪ Un gráfico con la representación de los diagramas de cajas o de bigotes de la colección de datos sintéticos y de la colección de datos reales tomados para calcular u. ▪ Un gráfico con la representación de distintos estadísticos (el rango y la mediana en especial) de la colección de datos sintéticos. ▪ Un gráfico con la representación de distintos estadísticos (el rango y la mediana en especial) de la colección de datos reales tomados para calcular u.
RF-15	El sistema deshabilitará la visualización de la colección y las evaluaciones.
RF-16	El sistema solicitará al usuario el nombre del archivo que contendrá la tabla.
RF-17	El sistema exportará una tabla con la colección de datos sintéticos en fotmato CSV y la almacenará en la carpeta del proyecto con el nombre introducido por el usuario. En caso de ausencia o invalidez del nombre, el sistema mostrará el mensaje "Por invalidez o ausencia de respuesta, se procede a usar el valor por defecto".
RF-18	El sistema exportará una imagen con una representación gráfica en fotmato PNG y la almacenará en la carpeta de descargas del ordenador con el nombre newplot.png.

Tabla 4.6: Requisitos funcionales

Requisitos no funcionales Se trata de requisitos sobre algunas restricciones y características que debe cumplir el sistema. Son muy importantes en el proceso de análisis de un proyecto *software* ya que conseguir un *software* de éxito no sólo requiere proveer la funcionalidad deseada (requisitos funcionales), sino que también importa cómo de bien se provee esa funcionalidad (atributos de calidad). Cabe mencionar que los atributos de calidad describen la mayoría de los requisitos no funcionales: seguridad, fiabilidad, rendimiento, eficiencia, reusabilidad... Sin embargo, las restricciones y los requisitos de interfaces externas también se consideran como requisitos no funcionales del sistema.

Atributos de calidad o requisitos de calidad Se establecen unos requisitos para facilitar la navegación del usuario, los cuales se describen en la Tabla 4.7 indicando, para cada uno, a qué tipo pertenecen.

ID	Tipo	Descripción
RNF-01	Rendimiento	El tiempo de respuesta del sistema dependerá del tamaño de los datos o número de registros $(L + 1)$, pero nunca será mayor de $0{,}2 * (L + 1)$ segundos.
RNF-02	Fiabilidad	El número máximo de experimentos que podrán fallar por motivos *software* es 5 de cada 1000.
RNF-03	Robustez	El sistema presentará una gran robustez frente a la ocurrencia de situaciones anómalas. Todos los parámetros que describen la serie temporal a generar tendrán un valor por defecto, de forma que el sistema utilizará estos en caso de que falte algún parámetro o su valor sea erróneo.
RNF-04	Usabilidad	El sistema será fácilmente usable. El 95 % de los usuarios que no hayan utilizado antes el sistema serán capaces de realizar una solicitud correcta de producto en menos de 10 minutos.
RNF-05	Reusabilidad	El código de las funciones será reutilizable en otros sistemas
RNF-06	Escalabilidad	La aplicación será compatible con los siguientes navegadores: *Microsoft Edge*, *Google Chrome*.
RNF-07	Accesibilidad	La aplicación se adaptará a diferentes tamaños de pantalla de ordenador: 640 X 480, 800 x 600, 2048 x 1536, 1024 x 768 píxeles.
RNF-08	Seguridad	El sistema deberá implementar mecanismos de bloqueo mediante *firewalls* para acceso no autorizado.

Tabla 4.7: Requisitos de calidad

Restricciones o reglas de negocio Las restricciones actúan como limitantes de las posibles decisiones que pueden tomarse en el diseño o la implementación del sistema a desarrollar. Se consideran las reglas de negocio que se observan en la Tabla 4.8.

ID	Tipo	Descripción
ReN-01	Restricción de implementación	El producto se implementará utilizando, exclusivamente, productos *open-source* disponibles bajo licencia *GNU-GPL*
ReN-02	Restricción de implementación	El código del producto se escribirá en Python

Tabla 4.8: Reglas de negocio

4.3. Diseño de *software*

A partir del modelo de análisis se deducen las estructuras de datos, la estructura en la que descompone el sistema y la interfaz de usuario. Como consecuencia de esta etapa se generan artefactos.

Los pasos que se realizan durante la etapa de diseño de cualquier aplicación *web* se desarrollan en diferentes subsecciones de la presente sección, y son:

- Diseño arquitectónico: Se identifica la estructura global del sistema y sus componentes, así como la relación existente entre ellos. Se corresponde con las subsecciones "Arquitectura lógica" (subsección 4.3.1) y "Arquitectura física" (subsección 4.3.2).
- Diseño de los componentes: Se especifica la forma en que debe funcionar cada componente, es decir, se describe la funcionalidad que debe implementarse. Se corresponde con el diagrama de clases de diseño y los diagramas de secuencia, que en este caso no se realizan.
- Diseño de datos: Se describe cómo implementar la base de datos. Se corresponde con la subsección "Modelo de datos" (subsección 4.3.3).
- Diseño de las interfaces: Se describen las interfaces gráficas de los componentes de forma ajena a su implementación. Se corresponde con el subsección "Diseño de interfaces de usuario" (subsección 4.3.4).

4.3.1. Arquitectura lógica

La aplicación se desarrollará en Dash, un *framework* de Python que está pensado para construir aplicaciones *web*, pero que se utiliza también para crear visualizaciones, porque permite personalizar *dashboards* o cuadros de mando.

Dash está basado, principalmente, en Flask (*framework* minimalista escrito en Python que permite crear aplicaciones *web* rápidamente y con un mínimo número de líneas de código [74]), Plotly [75] (proporciona herramientas de gráficos, análisis y estadísticas en línea para individuos y colaboración, así como bibliotecas de gráficos científicos para Python [76]) y ReactJS (biblioteca JavaScript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página [77]), y estas tres herramientas en las que se basa hacen que tenga las siguientes características:

- Las aplicaciones se reproducen en el navegador. Esto conlleva que una aplicación o servicio se puede desplegar en un servidor y, posteriormente, se puede compartir mediante Localizador Uniforme de Recursos o *Uniform Resource Locator* (URL).
- Es multiplataforma y está preparado para móviles, característica muy importante hoy en día. Además, tiene una librería de Bootstrap, con componentes de Bootstrap, que ayuda a ajustar correctamente el *layout* o diseño de la aplicación a cualquier pantalla en la que se quiera utilizar.
- Permite crear aplicaciones sencillas de manera muy rápida.
- Es *open-source*, es decir, es una herramienta de código abierto.

Las aplicaciones Dash se dividen en dos partes, una que se encarga del aspecto de la aplicación (*layout*) y otra de su interacción con el usuario (*callback*).

Dash describe el aspecto de la aplicación a través del *app.layout*, que consiste en un árbol jerárquico de componentes. Dash ofrece una serie de componentes para crear la interfaz de usuario. Estos componentes se dividen en dos paquetes: el paquete html o de *dash_html_component*,

donde se encuentran las etiquetas html y el paquete dcc o *dash_ core_ component*, que contiene los componentes interactivos que se encargan de la funcionalidad del tablero. Otro paquete utilizado en este sentido es el *dash_ bootstrap_ components*.

La interacción del tablero se controla mediante llamadas o *callbacks*. Es decir, en la capa de negocio se tienen *callbacks* o componentes de lógica de negocio para los elementos de interfaz, que son los que determinan el comportamiento de la aplicación frente a los eventos lanzados por el usuario. Dash permite definirlas para que al actualizar un componente del interfaz invoquen código Python capaz de realizar algún proceso y cambiar la interfaz.

El diagrama que representa la arquitectura lógica del sistema puede observarse en la Figura 4.3.

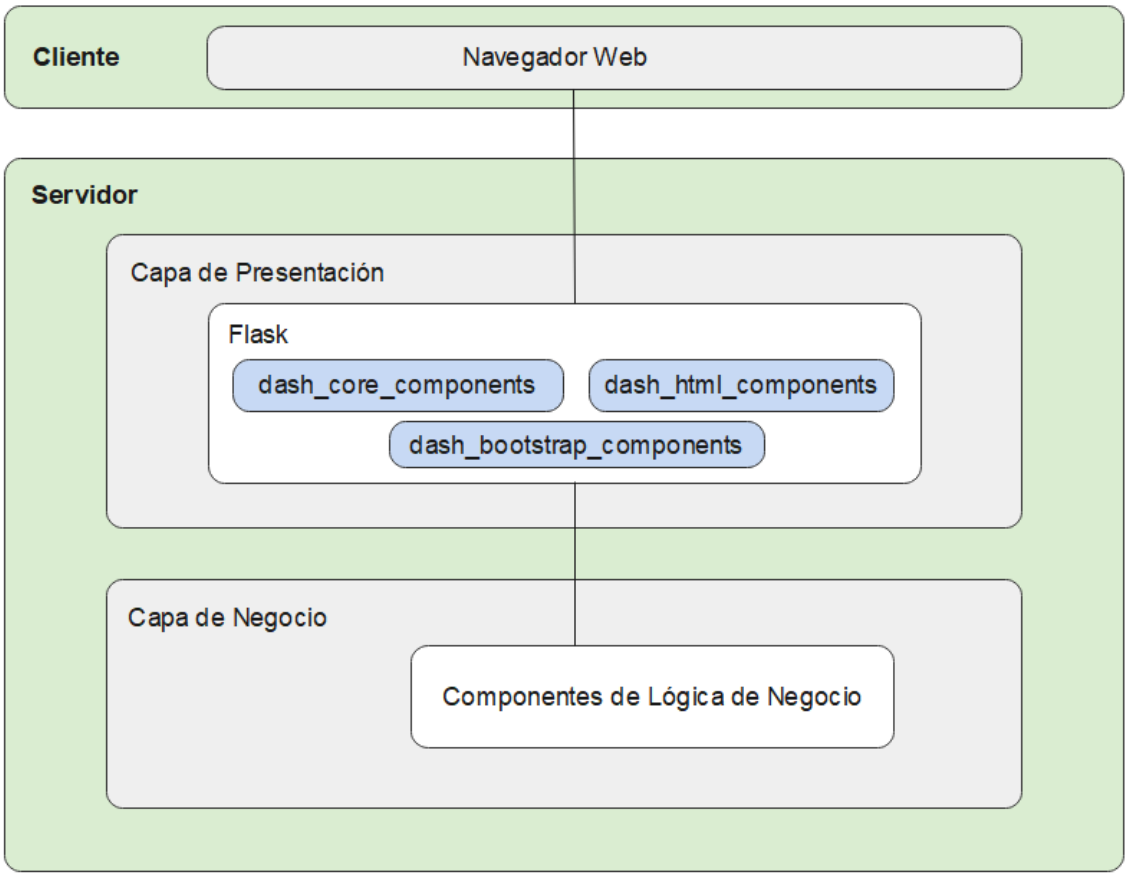


Figura 4.3: Arquitectura lógica

4.3.2. Arquitectura física

Puesto que toda arquitectura debe ser implementable en una arquitectura física, esta subsección se dedica a estudiarla. Se debe relacionar la arquitectura lógica con la física y crear un modelo de implantación física, en el que las capas (*layers*) de la primera se conviertan en capas (*tiers*) de la segunda. Por tanto, debido a esto y a que se pretende desarrollar una aplicación *web*

(cliente ligero), se opta por el modelo en 2 *tiers*.

La arquitectura en 2 niveles pasa la responsabilidad de la capa de presentación al cliente con respecto a las arquitecturas mononivel, ya obsoletas. Se conoce como Cliente/Servidor. Dependiendo de las responsabilidades del cliente se habla de clientes 'pesados' o 'ligeros'. De esta manera, se puede aprovechar la capacidad de cómputo del cliente, personalizar la capa de presentación para distintos fines y portarla a distintos entornos (multiplataforma) y mejorar la eficiencia en el lado del servidor [78].

Al igual que se estudian los requisitos funcionales para construir la arquitectura lógica, en el caso de la arquitectura física se deben tener en cuenta los requisitos no funcionales, ya estudiados en la subsección correspondiente del análisis. En consecuencia, a partir del RNF-08 (Tabla 4.7) de seguridad, es necesario implantar un *firewall* entre los *tiers*. Por su parte, el RNF-06 (Tabla 4.7) de escalabilidad hace necesario el uso de granjas de servidores de aplicaciones. Los balanceadores de carga ayudan en el proceso de escalar un sistema, por lo que se usan en la capa de aplicación. Además, aunque no aparece reflejado en ningún requisito no funcional específico, se requiere una alta disponibilidad de la aplicación (disposición del sistema para prestar servicio correctamente). Teniendo esto en cuenta, se ha construido la arquitectura física que se observa en la Figura 4.4.

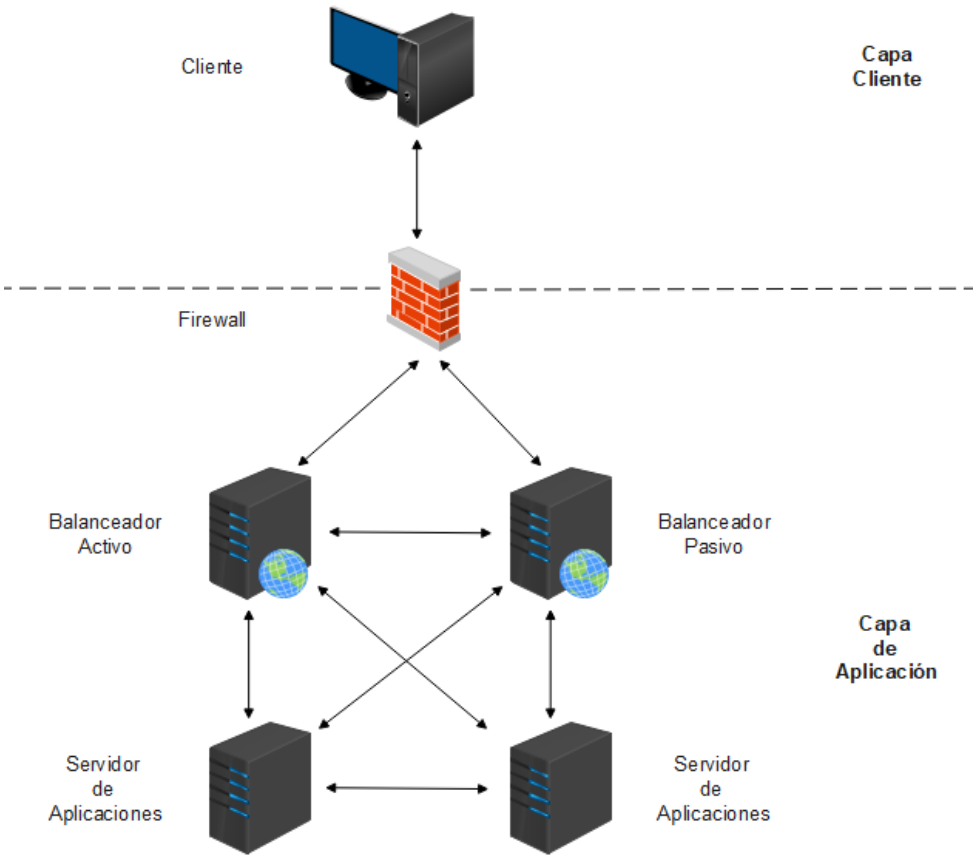


Figura 4.4: Arquitectura física

Diagrama de despliegue

En la Figura 4.5 se muestra el diagrama de despliegue del sistema. En este diagrama aparecen los dispositivos *hardware* («dispositivo») que componen la arquitectura física, y cada uno de ellos contiene un entorno de ejecución («entorno de ejecución»). Estos entornos se relacionan entre sí mediante los protocolos de comunicación especificados en el diagrama. Además, en cada servidor se despliega un artefacto, también presente en la figura.

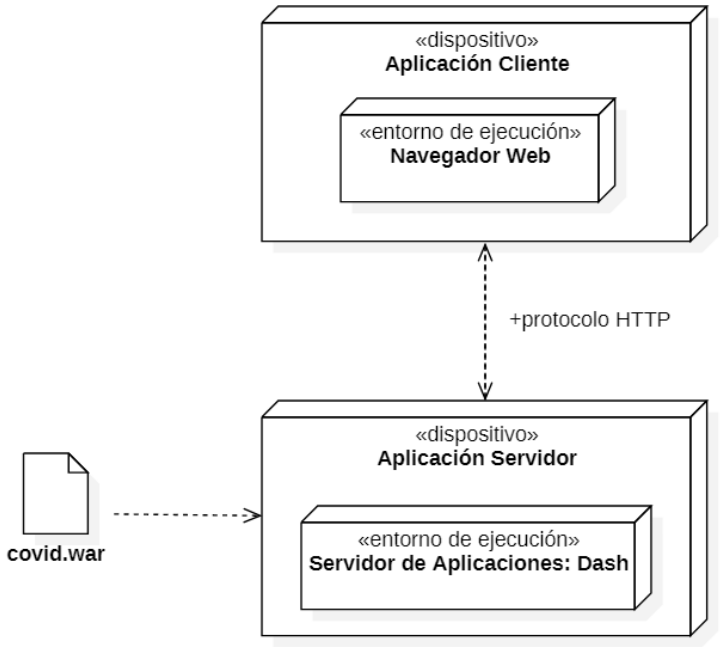


Figura 4.5: Diagrama de despliegue

4.3.3. Modelo de datos

El diccionario de datos completa la definición y descripción de los datos totalmente sintéticos que el sistema *software* de este proyecto genera, mostrando una visión mucho más detallada de su significado y posibles restricciones. Esta descripción facilita el trabajo al documentar de forma precisa qué son los datos y cómo se debe trabajar con ellos, lo que evita posibles descoordinaciones o inconsistencias en diferentes fases del modelado del sistema. Es por esto que resulta fundamental, junto con el modelo E-R, en el proceso de diseño conceptual de cualquier sistema *software*. Recoge tanto la definición de la entidad, posibles reglas y ejemplos de implementación (Tabla 4.9), como los atributos que posee y sus características: su definición, reglas de uso, si son multievaluados, si pueden ser nulos... (Tabla 4.10). Se especifica también cuál de sus atributos actúa como identificador (Tabla 4.11).

ID	Dato sintético COVID-19
Nombre	Dato sintético COVID-19
Definición	Un dato sintético COVID-19 es un registro que contiene información relevante acerca de la pandemia COVID-19 en una fecha y lugar concretos, como el número de personas susceptibles, expuestas, infectadas y recuperadas, en ese día y sitio fijados, de entre el número total de individuos de la población total del lugar, N. Ese lugar concreto tiene una política de restricciones relativas a la pandemia que su población debe acatar, pero dependiendo del miedo de sus habitantes y de los castigos impuestos frente a los negacionistas, los individuos obedecerán en mayor o menor medida. Todas estas cuestiones se parametrizan mediante las constantes β, γ y σ. El parámetro β es constante en caso de que no existan restricciones en el lugar, siendo una función definida a trozos en caso de existir tales restricciones. Dicha función está compuesta por dos trozos: uno constante e igual al parámetro β_0 y el otro una función dependiente del valor de β el día anterior. Se cambia de trozo en el momento $t = t_0$.
Notas	Registros de ejemplo: $[0, 99999, 0, 1, 0]$, $[1, 99567, 403, 30, 0]$, $[0, 99385, 507, 108, 0]$, $[0, 99311, 487, 202, 0]$.
Reglas	El valor de la suma de los atributos A02, A03, A04 y A05 ha de coincidir exactamente con N, el número total de individuos de la población.

Tabla 4.9: Diccionario de Datos - Entidad

ID	A01	A02	A03	A04	A05
Nombre	Día	Susceptibles	Expuestos	Infectados	Recuperados
Definición	Número que representa el orden en el que se encuentra el dato sintético dentro de la serie temporal.	Individuos susceptibles de ser infectados por el virus.	Individuos que portan la enfermedad pero que, al hallarse en su periodo de incubación, no muestran síntomas y aún no pueden infectar a otros.	Individuos ya infectados por el virus.	Individuos ya recuperados del virus. Entre ellos se encuentran también los individuos que fallecen.
Tipo	int	float	float	float	float
Reglas	Toma todos los valores entre 0 y L, ambos incluidos	$S + E + I + R = N$	$S + E + I + R = N$	$S + E + I + R = N$	$S + E + I + R = N$
Multivaluado	NO	NO	NO	NO	NO
Compuesto	NO	NO	NO	NO	NO
Único	SÍ	NO	NO	NO	NO
Inicial	NO	NO	NO	NO	NO
Nulo	NO	NO	NO	NO	NO

Tabla 4.10: Diccionario de Datos - Atributos

ID	A01
Nombre	Día

Tabla 4.11: Diccionario de Datos - Identificadores

4.3.4. Interfaces de usuario

Mediante el estudio del diseño de las interfaces de la aplicación, se pueden analizar formas en que el usuario interactúa con el sistema, hallando la manera más sencilla, efectiva, eficaz y satisfactoria posible de hacerlo en un contexto de uso definido. A continuación, se muestran cuatro diseños de interfaces de usuario de las múltiples que componen la aplicación y que se derivan de cada evento desencadenado por el usuario.

Si se observan los *mockups* con atención, se puede apreciar que, por motivos de accesibilidad (diseño Interfaz de Usuario o *User Interface* (IU) inclusivo) la pestaña o *tab* seleccionado tiene una apariencia distinta al resto para diferenciarlos. También es necesario darse cuanta de la distinta apariencia de las pestañas cuando están habilitadas y cuando no lo están.

En la Tabla 4.12 se muestra el diseño de la primera interfaz que observa el usuario al entrar en la aplicación.

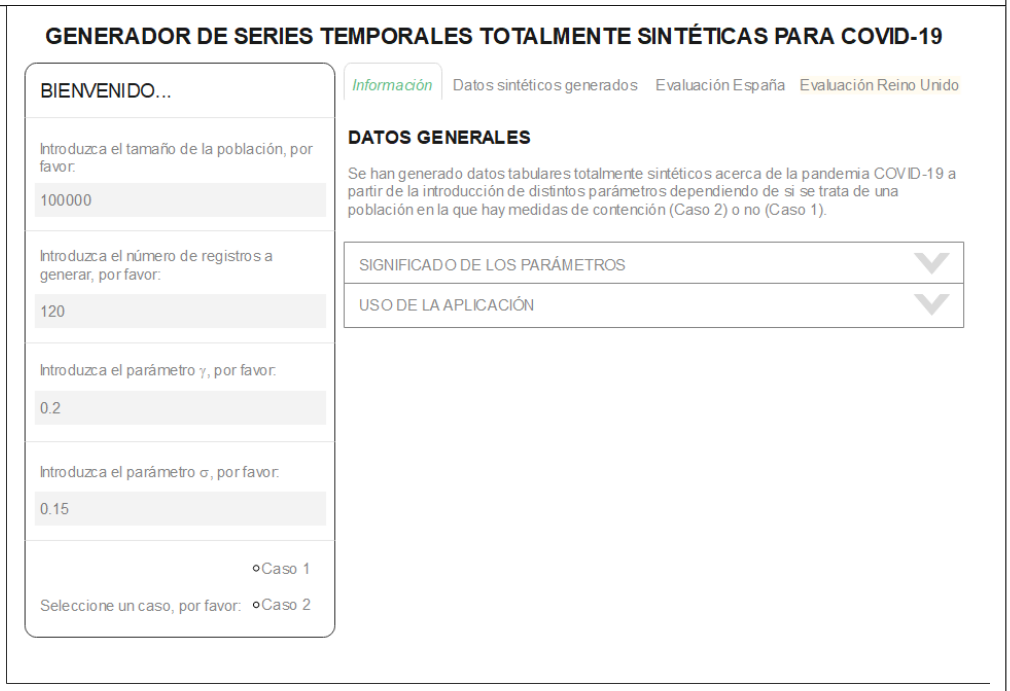
ID	IU-01
Descripción	Es la interfaz que aparece al entrar en la aplicación. En ella, se muestran el filtro, donde se solicitan los parámetros (comunes a ambos casos) y el caso, a la izquierda, y la pestañas a la derecha, siendo la pestaña "Información" la desplegada por defecto. El contenido de la pestaña, a la derecha, ofrece (en forma de acordeón) la posibilidad de visualizar el significado de los parámetros que caracterizan el modelo o un manual de uso de la aplicación.
Activación	El Usuario General inicia la aplicación.
Boceto	
Eventos	Iniciar la aplicación en el navegador.

Tabla 4.12: Interfaz 01

En la Tabla 4.13 se muestra el diseño de la interfaz de la aplicación que solicita al usuario los parámetros específicos del Caso 2.

ID	IU-02
Descripción	Es la interfaz que aparece cuando el usuario selecciona el Caso 2 tras entrar en la aplicación. En ella, se muestran el filtro a la izquierda, y la pestañas a la derecha, siendo la pestaña "Información" la desplegada por defecto. En el filtro se muestra un resumen editable de los parámetros (comunes a ambos casos) introducidos por el usuario, se solicitan los dos parámetros específicos del Caso 2, se muestra el caso seleccionado (con posibilidad de cambiarlo) y un botón para generar la colección de datos sintéticos. El contenido de la pestaña, a la derecha, ofrece (en forma de acordeón) la posibilidad de visualizar el significado de los parámetros que caracterizan el modelo o un manual de uso de la aplicación.
Activación	El Usuario General inicia la aplicación y selecciona el Caso 2
Boceto	
Eventos	Iniciar la aplicación. Seleccionar Caso 2.

Tabla 4.13: Interfaz 02

En la Tabla 4.14 se muestra el diseño de la interfaz de la aplicación que muestra al usuario la colección de datos generada a partir de los parámetros específicos del Caso 1 introducidos por el mismo y que también aparecen en forma de resumen editable en el filtro de la izquierda.

ID	IU-03
Descripción	Es la interfaz que aparece cuando el usuario pulsa el botón "GENERAR DATOS SINTÉTICOS" tras introducir los parámetros comunes y seleccionar el Caso 1. En ella, se muestran el filtro a la izquierda, y la pestañas a la derecha, siendo la pestaña "Datos sintéticos generados" la desplegada por selección del usuario. En el filtro se muestra un resumen editable de los parámetros (comunes a ambos casos) introducidos por el usuario, el caso seleccionado (con posibilidad de cambiarlo) y un botón para generar la colección de datos sintéticos. Debajo del botón una alerta indica que se ha generado la colección de manera satisfactoria (esta alerta durará solo unos segundos). El contenido de la pestaña, a la derecha, ofrece (en forma de acordeón) la posibilidad de visualizar la colección de forma tabular o gráfica.
Activación	El Usuario General inicia la aplicación, selecciona el Caso 1, pulsa el botón "GENERAR DATOS SINTÉTICOS" y presiona la pestaña "Datos sintéticos generados".
Boceto	
Eventos	Iniciar la aplicación. Seleccionar Caso 1. Pulsar botón "GENERAR DATOS SINTÉTICOS". Presionar la pestaña "Datos sintéticos generados".

Tabla 4.14: Interfaz 03

En la Tabla 4.15 se muestra el diseño de la interfaz de la aplicación que muestra al usuario la evaluación de la utilidad de la colección de datos generada (a partir de los parámetros específicos del Caso 2 introducidos por el mismo y que también aparecen en forma de resumen editable en el filtro de la izquierda) comparada con datos reales de Reino Unido.

ID	IU-04
Descripción	Es la interfaz que aparece cuando el usuario pulsa la pestaña "Evaluación Reino Unido" tras introducir los parámetros comunes, seleccionar el Caso 2, introducir los parámetros específicos de este caso, pulsar el botón "GENERAR DATOS SINTÉTICOS" y la pestaña "Datos sintéticos generados". En ella, se muestran el filtro a la izquierda, y la pestañas a la derecha, siendo la pestaña "Evaluación Reino Unido" la desplegada por selección del usuario. En el filtro se muestra un resumen editable de los parámetros (comunes a ambos casos y específicos del Caso 2) introducidos por el usuario, el caso seleccionado (con posibilidad de cambiarlo) y un botón para generar la colección de datos sintéticos. El contenido de la pestaña, a la derecha, ofrece (en forma de acordeón) la posibilidad de visualizar el resultado de una de las tres métricas de evaluación de la utilidad.
Activación	El Usuario General inicia la aplicación, introduce los parámetros comunes, selecciona el Caso 2, introduce los parámetros específicos de este caso, pulsa el botón "GENERAR DATOS SINTÉTICOS", la pestaña "Datos sintéticos generados" y la pestaña "Evaluación Reino Unido'.
Boceto	
Eventos	Iniciar la aplicación. Introducir los parámetros comunes. Seleccionar el Caso 2. Introducir los parámetros específicos de este caso. Pulsar el botón "GENERAR DATOS SINTÉTICOS". Presionar la pestaña "Datos sintéticos generados". Presionar la pestaña "Evaluación Reino Unido'.

Tabla 4.15: Interfaz 04

4.4. Implementación

En esta sección se detallan algunos de los aspectos claves de la implementación llevada a cabo a nivel técnico en el proyecto. En la subsección 4.4.1 se listan las herramientas y las tecnologías utilizadas durante esta fase del proyecto y en la subsección 4.4.2 se comenta que el código de la aplicación forma parte de la entrega del TFG.

4.4.1. Herramientas y tecnologías utilizadas

En esta subsección, se listan tanto las herramientas (tales como Entorno de Desarrollo Integrado o *Integrated Development Environment*s (IDEs)), como las tecnologías (lenguajes de programación, *frameworks* y librerías) utilizadas para la implementación del sistema *software*. Aquellas que no se detallan ya han sido comentadas con anterioridad.

Las herramientas que se han utilizado son:

- Anaconda: Distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos y ML [79].
- *JupyterNotebook*: Aplicación *web* de código abierto que permite crear y compartir documentos que integran código, resultados de ejecución y texto (en formato *Markdown*). Entre sus usos más extendidos destaca el procesado de datos, la simulación numérica, la creación de modelos estadísticos o de ML y la visualización de datos. Soporta más de 40 lenguajes de programación, incluidos Python y R [80]. *JupyterNotebook* ha sido utilizado durante el proyecto con Python para el procesado de datos y la construcción del generador de datos sintéticos.

Por otro lado, las tecnologías que se han empleado son:

- Python: Lenguaje de programación.
- Numpy: Librería de código abierto de Python.
- Scipy: Librería de código abierto de Python.
- Matplotlib: Librería de código abierto de Python.
- Pandas: Librería de código abierto de Python.
- Dash: Librería de código abierto de Python.
- Plotly: Librería de código abierto de Python.

A grandes rasgos, poder integrar la técnica de generación de datos sintéticos con aplicaciones *web*, de cara a la usabilidad futura por parte de personas ajenas al proyecto, ha sido lo que ha motivado la elección de Python como lenguaje para la implementación frente al lenguaje estadístico R. Además, la rápida curva de aprendizaje de Python y su versatilidad frente a R motiva a mayores su uso, dado que el objetivo del proyecto radica en resolver el problema a estudio, y el aprendizaje de las herramientas para resolverlo no debe convertirse en el núcleo del problema.

4.4.2. Aplicación *web*

En esta sección, se quiere recalcar que, junto con esta memoria, se adjunta el *notebook* que incluye el código comentado de la aplicación *web* construida.

El último bloque de código de este cuaderno permite ejecutar y desplegar dicha aplicación, lo que permite su prueba. A continuación, se muestra dicho bloque, formado por una línea de código comentada. Esta línea invoca la función de Dash que inicializa el servidor de aplicaciones *web* que implementa, despliega la aplicación y la sirve en un puerto local del ordenador para el acceso mediante el navegador. Además, se lanza el servidor con funciones de *debug*, para trazar cualquier error que pudiera producirse.

```
# Ejecucion y despliegue de la app
app.run_server(debug = True)
```

4.5. Manuales

En esta sección, se ofrecen dos manuales. El primero, comprende la serie de pasos que hay que dar para poder visualizar finalmente la aplicación en cualquier computador (subsección 4.5.1). El segundo, es una guía didáctica sencilla de la aplicación (subsección 4.5.2).

4.5.1. Manual de instalación

Para el desarrollo de esta aplicación se necesita únicamente Python, por lo que este manual es sencillo. Además, hay que disponer de la carpeta del proyecto, que consta de los archivos indicados en la Tabla 4.16. Los archivos `Spain.xlsx` y `UnitedKingdom.xlsx` se encuentran dentro de la carpeta `documents`, hija de la carpeta del proyecto.

Nombre	Formato	Contenido
`README.md`	*Markdown*	Manual de instalación o despliegue en inglés.
`requirements.txt`	Documento de texto	Versiones de las librerías y los *frameworks* que se necesitan. Se obtiene mediante los comandos `pip install` y `pip freeze`.
`Spain.xlsx`	Excell	Base de datos de España con la que se realizan las comparaciones.
`UnitedKingdom.xlsx`	Excell	Base de datos de Reino Unido con la que se realizan las comparaciones.
`generador.ipynb`	Documento *notebook*	*Notebook* de Python con todo el código de la aplicación.

Tabla 4.16: Archivos del proyecto

Toda la implementación del proyecto se han llevado a cabo utilizando Python como lenguaje de programación en su versión 3.9. En cuanto a las librerías requeridas, a continuación se listan las

versiones que han sido utilizadas durante el desarrollo del proyecto para las librerías y *frameworks* principales en orden alfabético, recogidas también en el archivo `requirements.txt` de la carpeta del proyecto. Algunas no aparecen debido a que ya vienen instaladas con Python.

- dash==2.6.1
- dash-bootstrap-components==1.2.1
- Flask==2.2.2
- jupyter-dash==0.4.2
- matplotlib==3.5.3
- numpy==1.23.3
- openpyxl==3.0.10
- pandas==1.4.4
- plotly==5.10.0
- scipy==1.9.1

Una vez conocida esta información, se explican en orden de realización, los pasos que hay que seguir tras conseguir la carpeta del proyecto.

1. Instalar Python 3.9 o versiones posteriores.
2. Comprobar si la ejecución de *scripts* está habilitada y, en caso contrario, habilitarla. Para ello, abrir el *Windows PowerShell* como administrador y ejecutar el comando `Get-ExecutionPolicy`. Si la salida es "Restricted" o, lo que es lo mismo, "Restringido", hay que cambiar esta configuración ejecutando `Set-ExecutionPolicy Unrestricted` e indicar "Si[S]". Este paso es necesario para evitar problemas durante el paso de activación del entorno virtual creado.
3. Instalar *Jupyter Notebook* (ejecutar el comando `pip install jupyter notebook` en el terminal) o, en su defecto, instalar *VSCode* junto con las extensiones de Python y *Jupyter Notebook*.
4. Abrir una de las herramientas del paso anterior.
5. Abrir su terminal en el directorio del proyecto.
6. Ejecutar el comando `pip install virtualenv` para disponer de dicha herramienta de creación de entornos Python virtuales.
7. Ejecutar el comando `virtualenv venv` para crear un entorno virtual con el que trabajar con las dependencias específicas de este proyecto y no ensuciar el entorno Python global.
8. Ejecutar el comando `./venv/Scripts/activate` para activar el entorno virtual creado en el paso anterior.

9. Ejecutar el comando `pip install -r requirements.txt` para instalar las dependencias específicas de este proyecto, recogidas en el archivo `requirements.txt`, en el entorno virtual activado en el paso anterior.

10. Abrir el archivo `generador.ipynb` y ejecutar todas sus celdas. La aplicación aparece en una nueva pestaña del navegador.

4.5.2. Manual de usuario

La interfaz gráfica de la aplicación *web* se ha diseñado para poder ser manejada sin necesidad de consultar ningún manual ni de requerir ninguna formación previa. De cualquier forma, en esta subsección se incluyen los conceptos básicos necesarios para utilizar la aplicación. Se aporta un manual de aprendizaje del sistema en el que se explica la realización de las distintas funcionalidades.

El usuario que acceda a la aplicación *web* podrá ver la pantalla de inicio, que tiene un título y está dividida en dos partes: un recuadro rectangular, a la izquierda, al que se va referir como filtro y un bloque de cuatro pestañas, a la derecha, con la primera pestaña seleccionada por defecto, ya que el resto están deshabilitadas. Las pestañas se van a nombrar según el número ordinal que corresponda con su lugar de aparición empezando por la izquierda. Esta pantalla se puede observar en la Figura 4.6.

Figura 4.6: Pantalla de inicio. Pestaña 1

La pestaña 1 contiene dos acordeones desplegables, cerrados siempre por defecto al seleccionar dicha pestaña. Si el usuario pulsa el primero, dicho acordeón se despliega mostrando el significado de los parámetros del modelo. Si el usuario pulsa el segundo acordeón, éste se despliega mostrando una guía de ayuda para el uso de la aplicación. Es importante saber que el despliegue paralelo de ambos acordeones es posible, es decir, la apertura de uno de ellos no significa el cierre del otro si ya estaba previamente abierto. Este despliegue simultáneo se puede observar en la Figura 4.7.

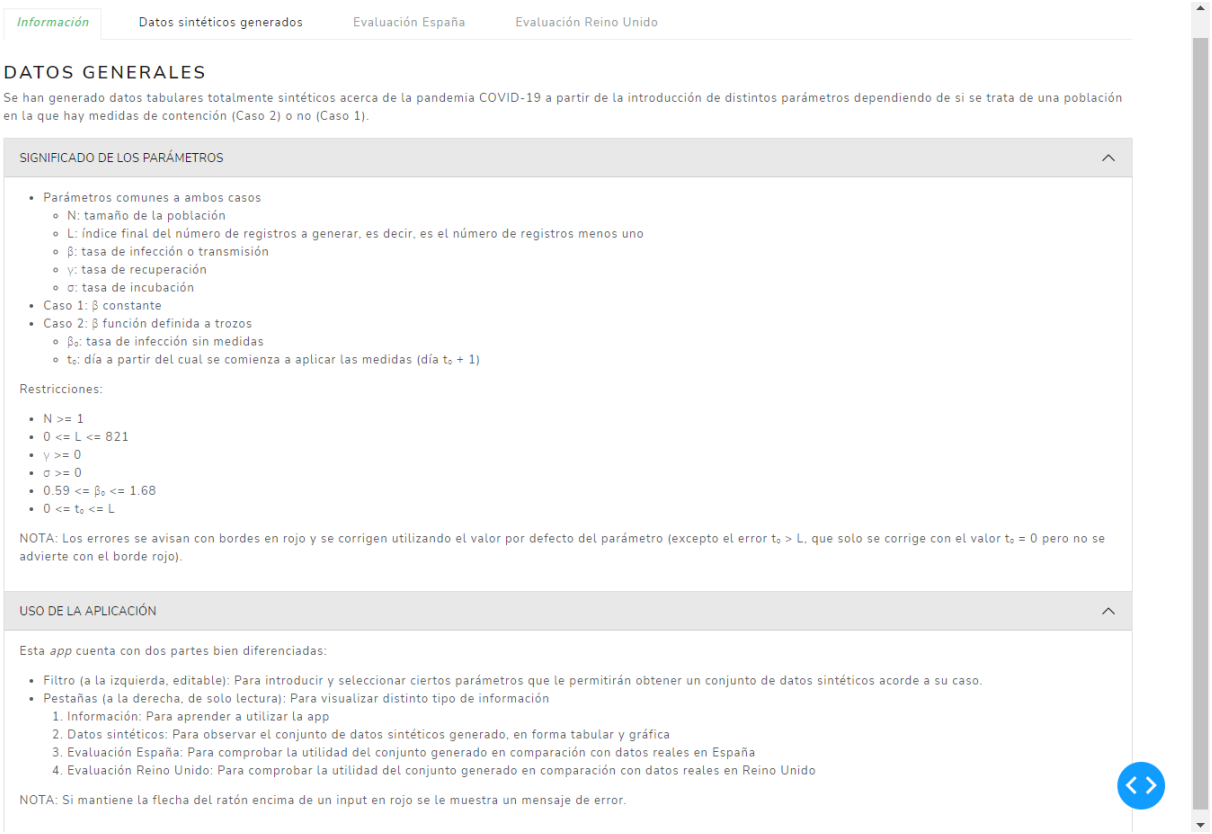


Figura 4.7: Pestaña 1. Significado de los parámetros y uso de la aplicación

El filtro contiene cuatro *inputs* o entradas, cada una rellena con un valor por defecto, que se corresponden con los parámetros comunes a ambos casos (explicados en la sección 4.1). Además, contiene un *radio item* que permite elegir un caso de los dos disponibles en el modelo SEIR. El usuario puede introducir un valor en cada uno de los *inputs* de dos maneras distintas: directamente por teclado o *clickando* en las flechas que aparecen a la derecha al pasar el ratón por encima del *input*. Todo esto se puede observar en la Figura 4.6.

Estas flechas permiten aumentar o disminuir el valor escrito en la caja en cada instante y, dependiendo del rango de valores permitido para cada entrada, avanza o desciende con un paso (*step*) más o menos grande, desde 0,001 para el parámetro β_0 hasta 1 para el resto de parámetros. Llegados a este punto resulta preciso comentar las restricciones a las que se ve sujeto cada uno de los parámetros:

- $1 \leq N$ y los saltos son de 1, es decir, una precisión máxima de unidades. La población ha de tener, al menos, un individuo y no puede tener un número decimal de ellos.
- $0 \leq L \leq 821$ y los saltos son de 1, es decir, una precisión máxima de unidades. Por tanto, dado que se generan $L+1$ registros, el tamaño permitido de la serie temporal se encuentra entre 1 y 822, debido a que las bases de datos reales con las que se compara registran los datos de la pandemia durante 822 días.
- $0 \leq \gamma$. La tasa de recuperación no puede ser un número negativo, pues supondría resultados de un número de personas por debajo de 0, lo cual es imposible.

- $0 \leq \sigma$. La tasa de incubación no puede ser un número negativo, pues supondría resultados de un número de personas por debajo de 0, lo cual es imposible.

- $0{,}59 \leq \beta_0 \leq 1{,}68$ y los saltos son de 0,001, es decir, una precisión máxima de milésimas. La tasa de infección antes de imponer medidas gubernamentales está restringida a estos valores como consecuencia del ya citado estudio [71]. La precisión se impone para que se pueda escoger más de un valor mediante las flechas.

- $0 \leq t_0 \leq L$ y los saltos son de 1, es decir, una precisión máxima de unidades. El día anterior a la entrada en vigor de las medidas gubernamentales no puede superar al número ordinal del último día de registros (L), ni mucho menos ser negativo o decimal.

Si el usuario elige el Caso 1, en el filtro se muestra un resumen editable de los valores de los parámetros (*inputs*), el caso seleccionado con posibilidad de cambio (*radio item*) y un botón, como se observa en la Figura 4.8.

Figura 4.8: Filtro (Caso 1)

Si el usuario elige el Caso 2, en el filtro se muestra un resumen editable de los valores de los parámetros (*inputs*), dos entradas adicionales para los parámetros específicos del Caso 2, el caso seleccionado con posibilidad de cambio (*radio item*) y un botón, como se observa en la Figura 4.9.

Figura 4.9: Filtro (Caso 2)

A partir de aquí, la explicación es común a ambos casos. Las entradas con valores inválidos se bordean en rojo y cuentan con mensajes informativos acerca del error si el usuario pasa el ratón por encima de ellas. Esto sucede con todas las entradas inválidas excepto para el caso $t_0 > L$ en la entrada de t_0, debido a problemas con el *kernel* en la implementación.

Si el usuario ha introducido algún valor nulo o inválido en los *inputs* de la pantalla de inicio, la caja correspondiente se bordea en rojo y, si el usuario deja el ratón encima, aparece un *tooltip* explicando la razón del error. Esto se observa en la Figura 4.10. Si el usuario deja dicho valor nulo o inválido y selecciona el caso, debajo de la caja correspondiente se le muestra una alerta roja (de aviso) de que se va a usar el valor por defecto su lugar, como se observa en la Figura 4.11.

Figura 4.10: Caja con borde rojo antes de seleccionar un caso en la pantalla de inicio

Figura 4.11: Alerta de aviso tras seleccionar un caso en la pantalla de inicio

Si el usuario ha introducido algún valor nulo o inválido en los *inputs* de la pantalla en la que el filtro muestra un resumen editable, la caja correspondiente se bordea en rojo y, si el usuario deja el ratón encima, aparece un *tooltip* explicando la razón del error. Esto se observa en la Figura 4.12. Si el usuario deja dicho valor nulo o inválido y pulsa el botón de generación, debajo de la caja correspondiente se le muestra una alerta roja (de aviso) de que se va a usar el valor por defecto su lugar, como se observa en la Figura 4.13.

Figura 4.12: Caja con borde rojo antes de dar al botón de generación

Figura 4.13: Alerta de aviso tras dar al botón de generación

Si el usuario pulsa el botón de generación, aparece una alerta verde (de éxito) que se mantiene durante 4 segundos; después, desaparece. Además, se habilita la pestaña 2, como se observa en la Figura 4.14.

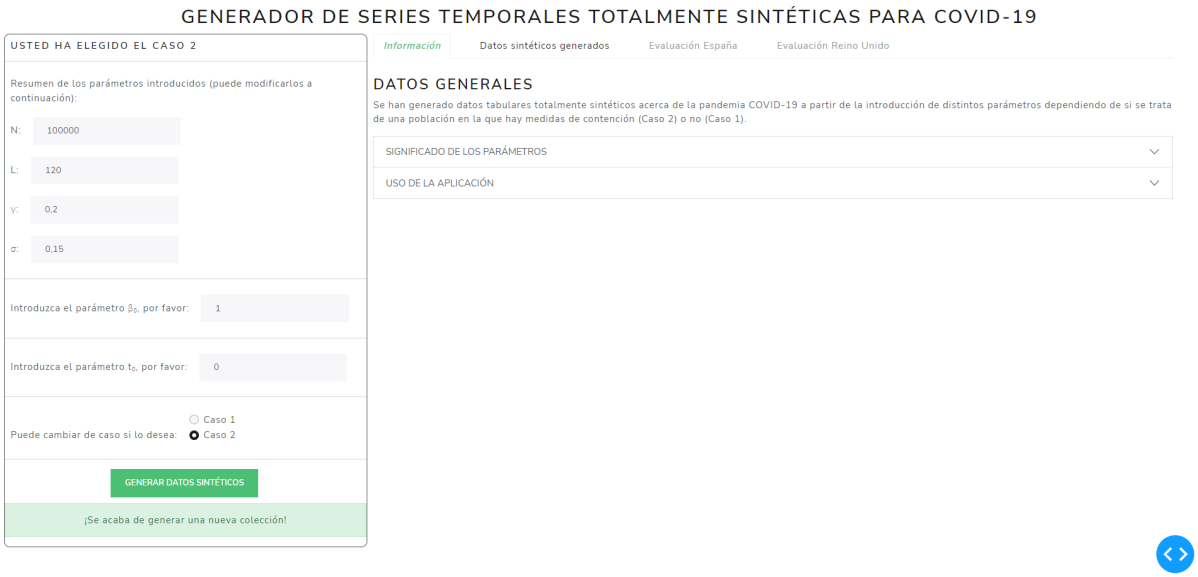


Figura 4.14: Alerta de éxito tras dar al botón de generación

Si el usuario presiona la pestaña 2, se muestra su contenido y se habilitan las pestañas 3 y 4. La pestaña 2 contiene dos acordeones desplegables, con posibilidad de apertura en paralelo, cerrados siempre por defecto al seleccionar dicha pestaña. Esto se observa en la Figura 4.15.

Figura 4.15: Pestaña 2

Si el usuario pulsa el primero, dicho acordeón se despliega mostrando la colección de datos sintéticos recién generada de forma tabular y la opción de descargar la tabla en formato CSV, pudiendo también introducir el nombre deseado para el archivo. Esta tabla divide sus filas en páginas de 10 y cada columna dispone de un filtro de coincidencias exactas. Si el usuario introduce un nombre para el archivo o no borra el mostrado por defecto y pulsa el botón de descarga, la

tabla se exporta en un archivo CSV, se guarda en la carpeta del proyecto y debajo del botón se le muestra una alerta verde (de éxito) que se mantiene durante 4 segundos; después, desaparece. Esto se observa en la Figura 4.16.

Figura 4.16: Pestaña 2. Representación tabular y mensaje de éxito tras pulsar el botón de descarga

Si el usuario ha introducido un valor nulo como nombre del archivo y pulsa el botón de descarga, debajo del botón se le muestra, además de la alerta verde ya citada, una alerta roja (de aviso) de que se va a usar el valor por defecto en lugar del nulo, como se observa en la Figura 4.17.

Figura 4.17: Pestaña 2. Representación tabular y mensaje de aviso tras pulsar el botón de descarga

Si el usuario pulsa el segundo acordeón, éste se despliega mostrando la colección de datos sintéticos recién generada de forma gráfica. Si el usuario pasa el ratón sobre la gráfica, aparecen varias opciones. Esto se puede observar en la Figura 4.18. La primera de todas es la más importante y la única que se va a explicar y *clickar* sobre ella permite al usuario descargar la gráfica en forma de imagen como archivo PNG. Se puede modificar el número de curvas que se muestran en la gráfica mediante la manipulación de la leyenda de ésta, es decir, si el usuario *clicka* sobre uno de los elementos de la leyenda, éste se oculta y vuelve a aparecer mediante el mismo proceso. Esta gráfica también muestra valores representados si el usuario pasa el ratón por encima.

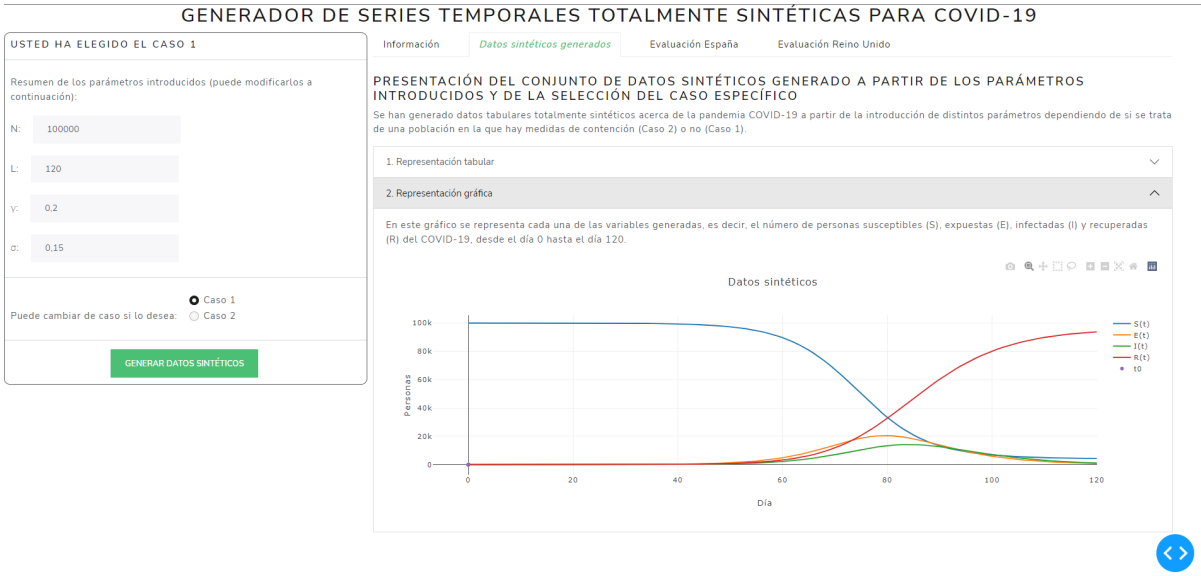


Figura 4.18: Pestaña 2. Representación gráfica

Si el usuario presiona la pestaña 3, se muestra su contenido. La pestaña 3 contiene tres acordeones desplegables, con posibilidad de apertura en paralelo, cerrados siempre por defecto al seleccionar dicha pestaña. Esto se observa en la Figura 4.19.

Figura 4.19: Pestaña 3

Si el usuario pulsa el primero, dicho acordeón se despliega mostrando el resultado de evaluar la utilidad de la colección de datos sintéticos recién generada mediante la métrica 1. Esto se observa en la Figura 4.20. Ídem para la pestaña 4.

Figura 4.20: Pestaña 3. Métrica 1

Si el usuario pulsa el segundo acordeón, éste se despliega mostrando el resultado de evaluar la utilidad de la colección de datos sintéticos recién generada mediante la métrica 2. Si el usuario pasa el ratón sobre la gráfica, aparecen varias opciones. Esto se puede observar en la Figura 4.21. La primera de todas es la más importante y la única que se va a explicar y *clickar* sobre ella permite al usuario descargar la gráfica en forma de imagen como archivo PNG. Se puede

modificar el número de curvas que se muestran en la gráfica mediante la manipulación de la leyenda de ésta, es decir, si el usuario *clicka* sobre uno de los elementos de la leyenda, éste se oculta y vuelve a aparecer mediante el mismo proceso. Esta gráfica también muestra valores representados si el usuario pasa el ratón por encima. Ídem para la pestaña 4.

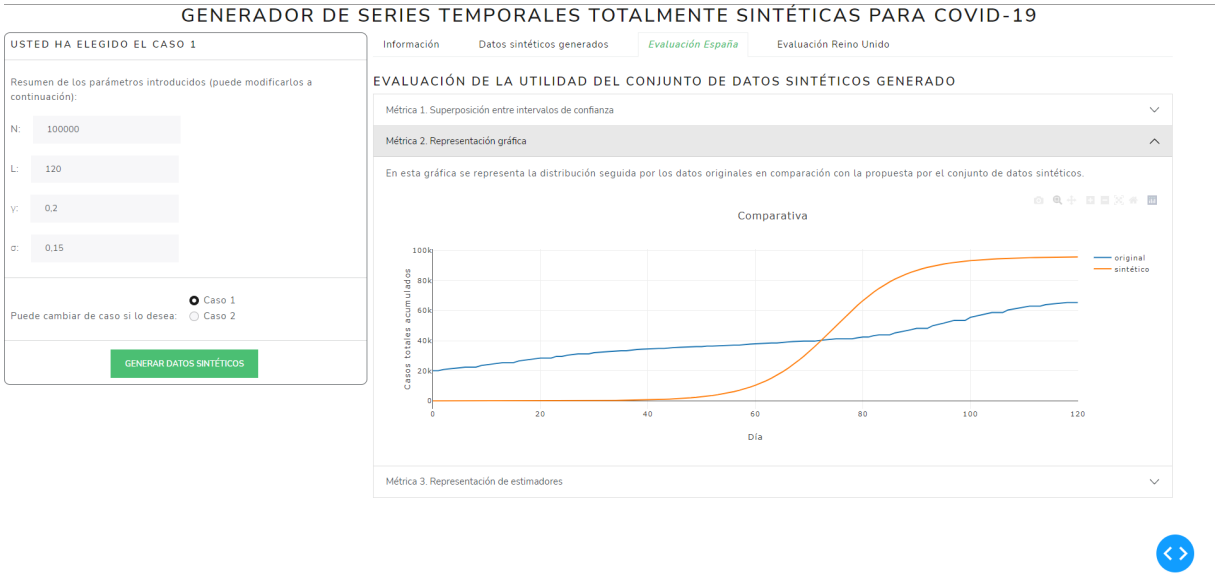


Figura 4.21: Pestaña 3. Métrica 2

Si el usuario pulsa el tercer acordeón, éste se despliega mostrando el resultado de evaluar la utilidad de la colección de datos sintéticos recién generada mediante la métrica 3, es decir, tres gráficas. Si el usuario pasa el ratón sobre cualquiera de las gráficas, aparecen varias opciones. Esto se puede observar en la Figura 4.22 y en la Figura 4.23. La primera de todas es la más importante y la única que se va a explicar y *clickar* sobre ella permite al usuario descargar la gráfica en forma de imagen como archivo PNG. Se puede modificar el número de estimadores que se muestran en la gráfica mediante la manipulación de la leyenda de ésta, es decir, si el usuario *clicka* sobre uno de los elementos de la leyenda, éste se oculta y vuelve a aparecer mediante el mismo proceso. Estas gráficas también muestran valores representados si el usuario pasa el ratón por encima. Ídem para la pestaña 4.

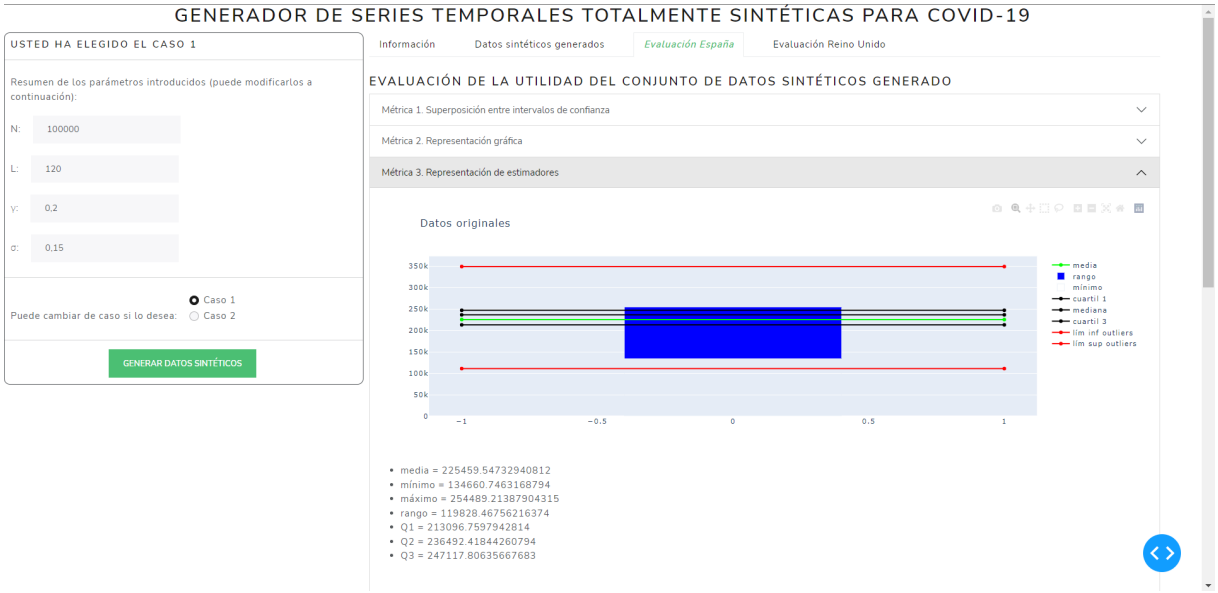


Figura 4.22: Pestaña 3. Métrica 3.1

Figura 4.23: Pestaña 3. Métrica 3.2. Métrica 3.3

Si el usuario cambia de caso con el *radio item* reservado para tal acción en el filtro, se deshabilitan las pestañas 2, 3 y 4 y se muestra el contenido de la pestaña 1. Sin embargo, si el usuario cambia cualquiera de los demás valores editables del filtro, no sucede nada hasta que el usuario vuelva a pulsar el botón de generar, que deshabilita las pestañas 3 y 4 y muestra el contenido de la pestaña 1.

Capítulo 5

Experimentación y evaluación

Este capítulo tiene como objetivo principal experimentar con el sistema *software*, probarlo, evaluarlo de acuerdo a ciertas métricas y poder así validar que satisface los objetivos de este proyecto.

5.1. Diseño experimental

En esta sección, se comentan tanto las métricas utilizadas para evaluar la colección de datos sintéticos generada (subsección 5.1.1) como la fuente y el significado del conjunto datos de prueba con el que se compara, a los que también se hace referencia como datos originales o reales (subsección 5.1.2).

5.1.1. Métricas

En el caso de este sistema, se evalúa la utilidad del conjunto de datos generado mediante el uso de tres métricas distintas. La primera de ellas es completamente cuantitativa y ya ha sido comentada en la subsección 3.3.2, mientras que las dos siguientes se expresan de forma gráfica.

Métrica 1. Superposición entre intervalos de confianza

Mediante esta primera métrica, se evalúa la utilidad de los datos generados midiendo la diferencia entre la calidad de la estimación de un determinado estimador para datos sintéticos y datos originales. Para ello, se usa un mecanismo basado en la superposición entre intervalos de confianza para evaluar la efectividad de estimaciones específicas.

Sea $x \in X$ los casos totales diarios acumulados que se generan sintéticamente, es decir, la suma de las personas expuestas, infectadas y recuperadas. Luego $X(t) = E(t)+I(t)+R(t), \forall t$. Se estima la media de x, μ (media poblacional), usando el estimador puntual, $\overline{x}$ (media muestral), y se construye un intervalo de confianza del 90 % alrededor del estimador como se indica en la Ecuación 5.1. En ella, $z_{\alpha/2}$ es el valor crítico de la distribución normal estandarizada, σ es la desviación estándar muestral y N es el número de individuos o tamaño de la muestra.

$$\text{Intervalo de confianza} = \left[\overline{x} \mp z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{N}}\right] \tag{5.1}$$

En general, el nivel de confianza se simboliza con $(1-\alpha)\cdot 100\,\%$, donde α es la proporción de las colas de la distribución que está fuera del intervalo de confianza. La proporción de la cola superior e inferior de la distribución es $\alpha/2$. Para un intervalo del 90 % de confianza, $(1-\alpha)\cdot 100\,\% = 90\,\%$, dicha proporción se calcula como en 5.2.

$$\alpha/2 = \frac{1-\frac{90}{100}}{2} = \frac{1-0{,}9}{2} = \frac{0{,}1}{2} = 0{,}05 \tag{5.2}$$

Se llama valor crítico al valor de $z_{\alpha/2}$ necesario para construir un intervalo de confianza para la distribución [81]. El valor crítico $z_{\alpha/2}$ correspondiente al área acumulativa de 0,9 es 1,645, porque hay 0,05 en la cola superior de la distribución y el área acumulativa menor a $z_{\alpha/2} = 1{,}645$ es 0,95.

Sea (L_s, U_s) el intervalo del 90 % de confianza de los datos generados sintéticamente y (L_o, U_o) el intervalo del 90 % de confianza de los datos originales, se calcula la intersección de estos intervalos y se denota como (L_i, U_i). La medida de utilidad de superposición se calcula como se indica en la Ecuación 5.3. El valor de u es cercano a 1 si se preserva la utilidad y $u = 0$ se refiere a la similitud nula de los intervalos de confianza.

$$\text{Medida de utilidad (u)} = \frac{(U_i - L_i)}{2\cdot(U_o - L_o)} + \frac{(U_i - L_i)}{2\cdot(U_s - L_s)} \tag{5.3}$$

Con el fin de conseguir la mejor de las evaluaciones, se calculan todas las intersecciones y se toma la que genera un valor de u más alto. Para ello, se utiliza un bucle que va recorriendo la Base de Datos o *Database* (BD) (archivo CSV) correspondiente desde el principio en bloques de $L + 1$ registros y calculando el intervalo (L_o, U_o) para ese bloque. La intersección de dicho intervalo con (L_s, U_s) se guarda en un array, en la posición correspondiente al contador del bucle menos dos unidades (desfase debido a que el primer registro de la BD comienza en la segunda celda). En este proceso, el conjunto de datos sintéticos generado se mantiene fijo y el intervalo correspondiente a los datos sintéticos se realiza sobre la suma de los expuestos, los infectados y los recuperados, esto es, para $E + I + R$ sintéticos. Esta es la razón por la que el tiempo de ejecución del programa al evaluar es algo alto, ya que ha de recorrer el conjunto de datos reales varias veces en busca de la mejor comparación.

Métrica 2. Representación gráfica

Esta segunda métrica trata de ofrecer al usuario una comparativa totalmente visual de las curvas de los casos totales acumulados reales y sintéticos. Para ello, se representan, sobre el mismo eje de coordenadas, el total de casos acumulados reales escalado a una población de N individuos y el total de casos acumulados generados sintéticamente. El escalado se realiza mediante una sencilla regla de 3.

Cabe mencionar que los registros del conjunto de datos reales que se toman en esta métrica dependen del resultado de la métrica 1 (sección 5.1.1), tomándose aquellos con los que se obtiene la medida de utilidad (u) más alta.

Métrica 3. Representación de estimadores

Esta tercera métrica se centra en el estudio de estimadores de ambos conjuntos. Por ello, conviene hacer un repaso de esta parte de la estadística [82].

Los estadísticos son resúmenes de los datos muestrales. Describen una distribución según cómo se comporta el centro, su dispersión y su forma. Se agrupan en estadísticos de tendencia central, posición, dispersión y forma. Por un lado, los estadísticos de tendencia central se ubican al centro de la distribución de los datos y un ejemplo de este tipo de estadísticos es la mediana, que es el valor central en el 50 %. La **mediana** es un estadístico robusto; aunque los extremos de los datos se vean alterados, la mediana permanece invariable. La **media** aritmética es el centro de gravedad de los datos y la **moda**, por su parte, es el valor de la variable que se repite con mayor frecuencia.

Existe una estrecha relación entre los tres principales estadísticos de tendencia central que permite clasificar las distribuciones según su simetría. Si la media, la moda y la mediana coinciden, se dice que la distribución es simétrica. Si la media es mayor que la mediana, se dice que la distribución es asimétrica con cola a la derecha (sesgada a la derecha). Por último, si la media es menor que la mediana, se dice que la distribución es asimétrica con cola a la izquierda (sesgada a la izquierda). Por otro lado, los estadísticos de posición son valores de la variable que dividen a la muestra en partes de igual porcentaje. Los **cuartiles** separan la muestra en grupos de 25 % cada uno y son 3. El segundo cuartil coincide con la mediana.

Las medidas de tendencia central son útiles pero dan una interpretación parcial de los datos, por lo que se hace necesario el estudio de estadísticos de dispersión. El **rango** es la medida de variabilidad o dispersión más simple que existe y se calcula tomando la diferencia entre el valor máximo y el mínimo observados. La diferencia entre el tercer cuartil y el primer cuartil se llama **rango entre cuartiles**, que mide la variabilidad de la mitad central de los datos. La **desviación estándar** es una medida de la dispersión de las observaciones a la media y se calcula como un promedio de la distancia de las observaciones a la media. La desviación estándar es positiva y es mayor cuanto más alejados estén los valores del promedio. Así como el promedio es una medida de tendencia central que no es resistente a las observaciones extremas, la desviación estándar, que usa el promedio en su definición, tampoco es una medida de dispersión resistente a valores extremos. Estos tres estadísticos son medidas de la variabilidad de un conjunto de datos.

Cuando se quiere describir una variable, se usa una medida de posición central y una medida de dispersión. El par de medidas más comúnmente usado es la media aritmética y la desviación estándar. Sin embargo, cuando la distribución de las observaciones es sesgada, la media no es una buena medida de posición central y es preferible usar la mediana. La mediana, en general, va acompañada del rango como medida de dispersión; aún así, cuando se observan valores extraños (extremos), el rango se ve muy afectado, por lo que es preferible usar el rango entre cuartiles o intercuartílico.

También es importante recordar lo que son los valores extremos o anómalos (*outliers*), que son observaciones que se alejan del conjunto de datos. Los valores extremos, por lo general, son atribuibles a que la observación se registra incorrectamente, a que proviene de una población distinta o a que es correcta pero representa un suceso poco común o fortuito. Una regla para determinar si un dato es *outlier* es la siguiente:

- Si un dato es $< Q_1 - 3 \cdot (Q_3 - Q_1)$
- Si un dato es $> Q_3 + 3 \cdot (Q_3 - Q_1)$

Por ello, se representan, de forma vertical y en ejes coordenados separados para cada uno de los conjuntos de valores reales y sintéticos, la media, el mínimo, el máximo (y, por consiguiente, el rango), los tres cuartiles y los límites a partir de los cuales un dato es considerado *outlier*. En

un principio solo se pretendía representar el rango y la mediana pero se ha preferido añadir el resto de valores para que el usuario tenga toda la información posible.

Por último, se representa de forma horizontal y en un mismo eje de coordenadas, el diagrama de caja y bigotes de cada colección de datos, incluyendo mediante puntos la representación de todos los valores aunque puedan ser *outliers*. De esta manera, se pueden comparar ambas representaciones y sacar numerosos resultados. Por ejemplo, se puede estudiar si tienen la misma simetría, si presentan *outliers*...

Cabe mencionar que los registros del conjunto de datos reales que se toman en esta métrica dependen del resultado de la métrica 1 (sección 5.1.1), tomándose aquellos con los que se obtiene la medida de utilidad (u) más alta.

5.1.2. Datos de prueba

El conjunto de datos de prueba consiste en dos bases de datos extraídas de la página *web* de Radio TV Española (RTVE) [83]. Se trata de dos bases de datos con las mismas columnas: fecha, casos totales acumulados, nuevos casos, *tests* totales, ratio positiva, muertes totales, personas vacunadas y población, seleccionadas de entre otras muchas variables que recoge dicha organización. Estas bases de datos contienen registros referentes a España y Reino Unido, respectivamente, desde el 01/02/2020 (comienzo de la extensión mundial del coronavirus) hasta el 02/05/2022, ambos inclusive. La BD se actualiza cada día y contiene datos de países de todo el mundo. La elección de estos dos países es debida a que son los países utilizados para el TFG de Matemáticas del estudiante autor del presente trabajo.

Dado que la segunda columna trata una acumulación de casos, se tienen en cuenta tanto los infectados (personas propagadoras del virus o no) como los recuperados (personas que han superado el virus satisfactoriamente o lamentablemente fallecidas). Por esta razón, cuando se compara esta información real con el caso sintético, se suma el número de personas expuestas, infectadas y recuperadas.

5.2. Experimentación y resultados

Este proceso de evaluación a partir de las métricas descritas en la subsección 5.1.1 es visible por el propio usuario de la aplicación y se encuentra disponible en las pestañas tituladas "Evaluación España" y "Evaluación Reino Unido", comparándose la colección sintética con un subconjunto de la colección real de España y Reino Unido, respectivamente.

En esta sección se van a mostrar, con su consiguiente explicación, los resultados obtenidos con cada una de las métricas en tres ejemplos distintos. El procedimiento seguido con cada uno de los ejemplos va a ser el mismo; primero, se citan los valores de cada uno de los parámetros, el caso escogido y se indica a qué país pertenece la comparación; después, se adjunta una captura de los resultados obtenidos con cada una de las métricas; y, finalmente, se comentan las conclusiones derivadas de los resultados.

Cabe comentar que, aunque en ciertos gráficos no se distinguen los cuartiles o los límites establecidos para los *outliers* debido a superposición de líneas, es posible visualizar cada elemento del gráfico con normalidad gracias a que, *clickando* sobre su nombre en la leyenda, se pueden ocultar y volver a dibujar las veces que se desee.

Ejemplo 1 A continuación, se listan los valores de cada uno de los parámetros, el caso escogido y se indica a qué país pertenece la comparación.

- Tamaño de la población: $N = 100000$
- Número de registros a generar menos uno: $L = 120$
- Tasa de recuperación: $\gamma = \frac{1}{5} = 0{,}2$
- Tasa de incubación: $\sigma = \frac{1}{7} = 0{,}15$
- Caso = 1 (Sin medidas gubernamentales de contención)
- País = España

En la Figura 5.1 se puede observar la salida de la métrica 1. Como comenta la última frase de la imagen, se obtiene una superposición "óptima" más cercana a 1 que a 0, por lo que la colección generada es válida en cuanto a utilidad retenida se refiere.

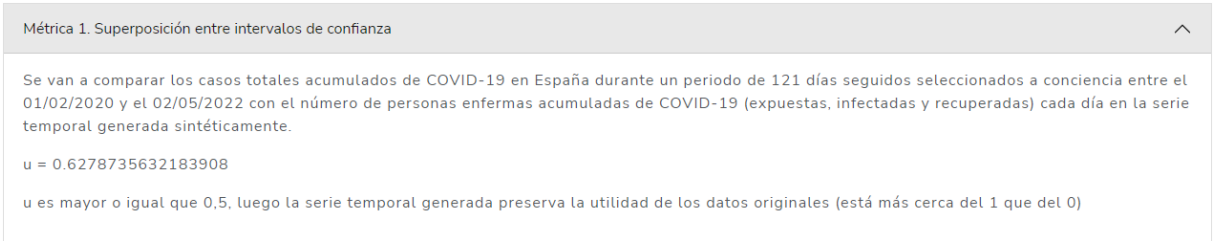


Figura 5.1: Ejemplo 1. Métrica 1

En la Figura 5.2 se puede observar la salida de la métrica 2. En esta gráfica se ve que ambas curvas son crecientes (salvo excepciones puntuales en el caso de los datos originales) y que, mientras que una es suave, la otra es pronunciada. La interpretación de las curvas resulta en que, en el caso sintético, aproximadamente el día 40 se produce un "pico" de casos totales acumulados mientras que durante el resto del periodo registrado se produce un estancamiento (la función es aproximadamente constante en los intervalos $[0, 35]$ y $[45, 120]$). En el caso original, el aumento de casos totales acumulados es progresivo. Las causas de esta diferencia se explican en el segundo párrafo de la subsección 5.2.

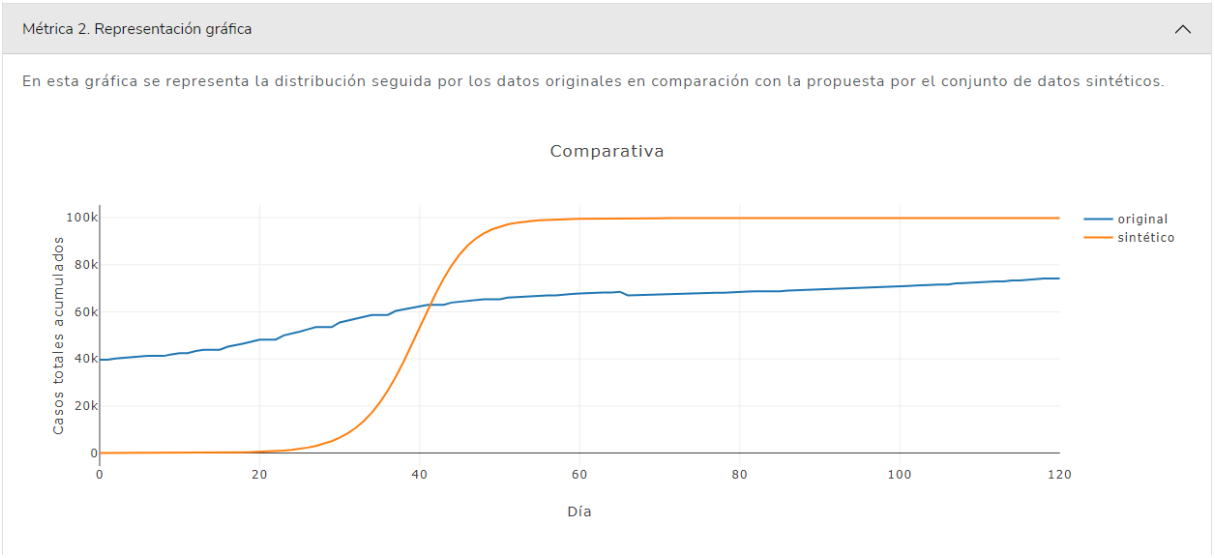


Figura 5.2: Ejemplo 1. Métrica 2

En las Figuras 5.3, 5.4 y 5.5 se puede observar la salida de la métrica 3. De las primeras dos imágenes se puede concluir que ambas colecciones tienen la misma simetría, pues en ambas la media es menor que la mediana o cuartil 2. Concretamente, son distribuciones asimétricas con cola a la izquierda (sesgadas a la izquierda). No obstante, estas imágenes también expresan que ninguna de las colecciones presenta *outliers* (la barra azul se encuentra entre las líneas rojas en ambos conjuntos).

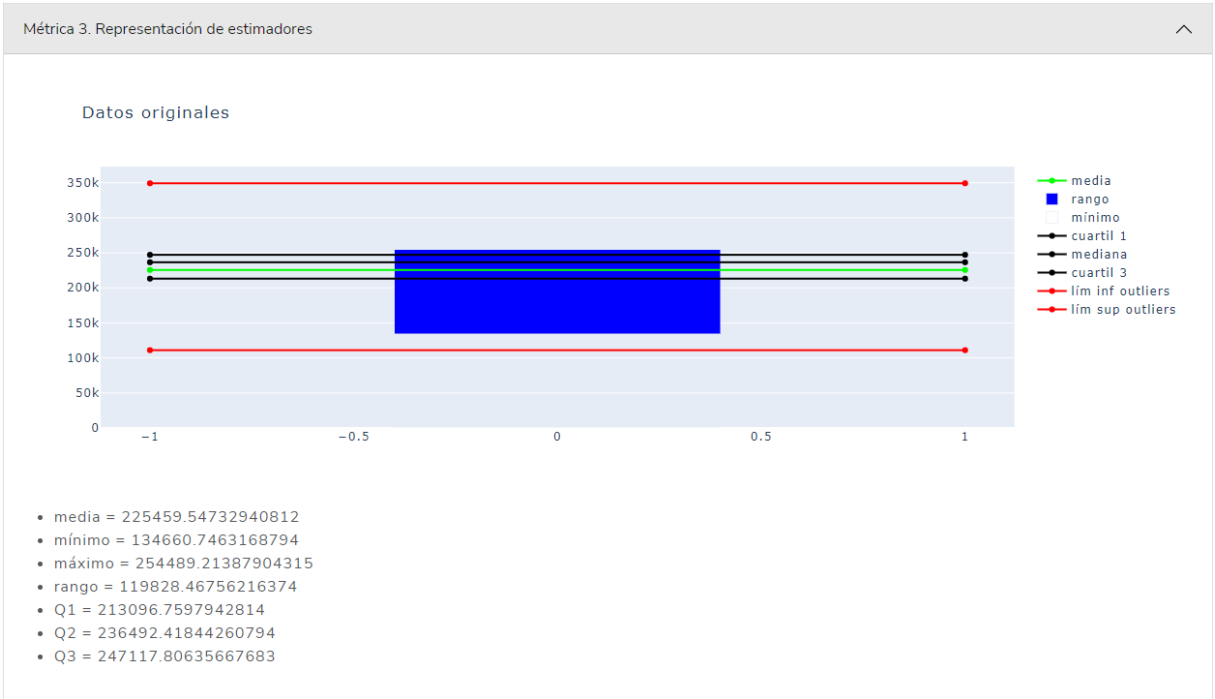


Figura 5.3: Ejemplo 1. Métrica 3.1

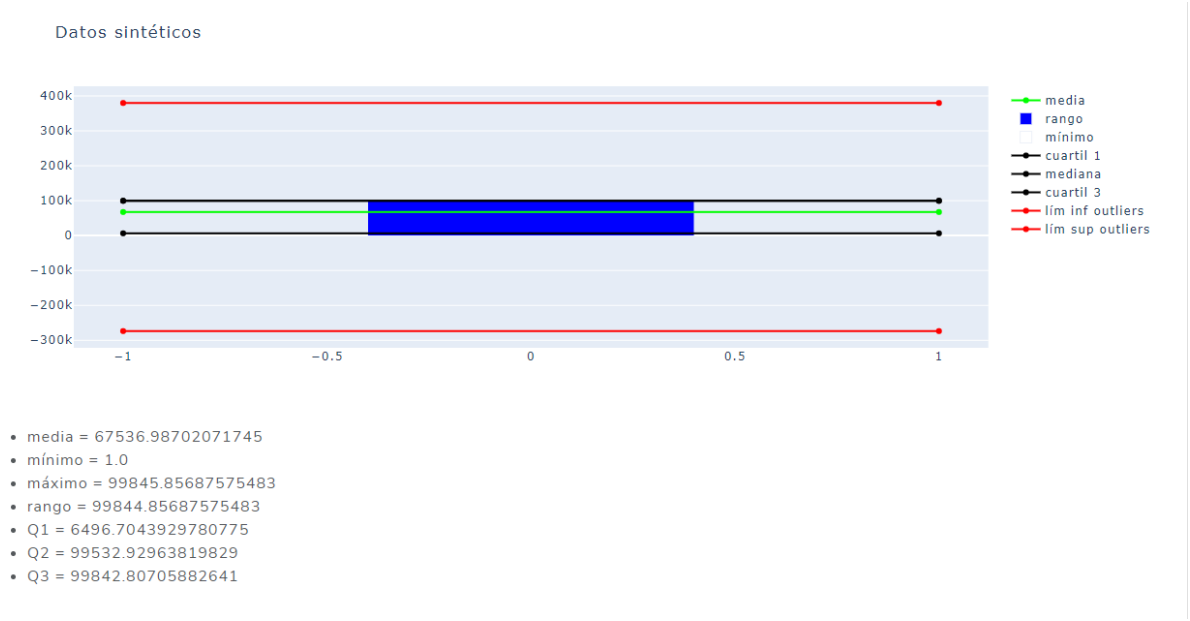


Figura 5.4: Ejemplo 1. Métrica 3.2

La Figura 5.5 corrobora la ausencia de *outliers* en ambos conjuntos de datos, ya que todos los puntos se sitúan entre los bigotes de cada gráfica en cada caso. A pesar de que en estos diagramas no se representa la media, se observa que la mediana de ambos conjuntos no está en el centro de la caja, de donde se deduce que ninguna de las muestras es simétrica. Que la parte izquierda de la caja sea mayor que la de la derecha en ambos casos, quiere decir que los casos totales acumulados originales y sintéticos comprendidos entre el 25 % y el 50 % de la población están más dispersos que entre el 50 % y el 75 %. Esto puede observarse también en la concentración de los puntos representados debajo de cada caja. El bigote de la izquierda es más corto que el de la derecha en el caso de los datos sintéticos, por lo que el 25 % de los días con menos casos totales acumulados registrados están más concentrados que el 25 % de los días con más casos totales acumulados registrados. El bigote de la izquierda es más largo que el de la derecha en el caso de los datos sintéticos, por lo que el 25 % de los días con más casos totales acumulados registrados están más concentrados que el 25 % de los días con menos casos totales acumulados registrados. El rango intercuartílico, es decir, el 50 % de la población está comprendido en 15000 casos y 90000 casos, para el conjunto de datos original y sintético, respectivamente [84].

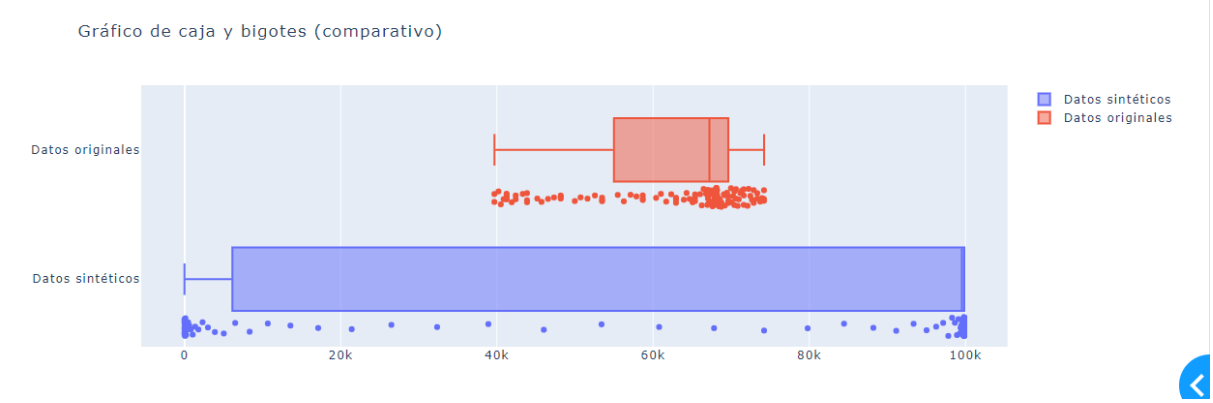


Figura 5.5: Ejemplo 1. Métrica 3.3

Ejemplo 2 A continuación, se listan los valores de cada uno de los parámetros, el caso escogido y se indica a qué país pertenece la comparación. Este ejemplo se lleva a cabo con el fin de poner de manifiesto que una tasa de incubación mayor que la de recuperación da lugar a valores extraños o *outliers*, mientras que, el caso contrario, representado en el ejemplo siguiente, lleva a buenos resultados.

- Tamaño de la población: $N = 5000$
- Número de registros a generar menos uno: $L = 240$
- Tasa de recuperación: $\gamma = 0{,}1$
- Tasa de incubación: $\sigma = 0{,}3$
- Caso = 2 (Con medidas gubernamentales de contención)
- Tasa de infección antes de aplicar las medidas: $\beta_0 = 1$
- Día a partir del cual se comienza a aplicar las medidas (día $t_0 + 1$): $t_0 = 20$
- País = Reino Unido

En la Figura 5.6 se puede observar la salida de la métrica 1. Como comenta la última frase de la imagen, se obtiene una superposición "óptima" más cercana a 1 que a 0, por lo que la colección generada es válida en cuanto a utilidad retenida se refiere. De hecho, en este caso roza la perfección de la superposición.

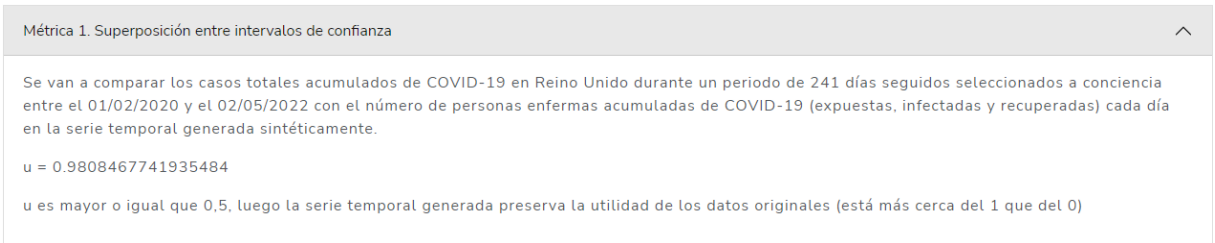


Figura 5.6: Ejemplo 2. Métrica 1

En la Figura 5.7 se puede observar la salida de la métrica 2. En esta gráfica se ve que ambas curvas son crecientes y que, mientras que una es suave, la otra es pronunciada. En la colección de datos sintéticos existe un conjunto de días en los que los casos suben muy rápidamente mientras que en los datos reales la subida es progresiva y lenta. La interpretación de las curvas resulta en que, en el caso sintético, aproximadamente el día 25 se produce un "pico" de casos totales acumulados mientras que durante el resto del periodo registrado se produce un estancamiento (la función es aproximadamente constante en los intervalos $[0, 35]$ y $[45, 120]$). En el caso original, existe un primer periodo en el que prácticamente hay constancia en el número de casos totales acumulados registrados, es decir, no se producen nuevos casos; mientras que, aproximadamente a partir del día 10, el aumento de casos totales acumulados es progresivo. Las causas de esta diferencia se explican en el segundo párrafo de la subsección 5.2.

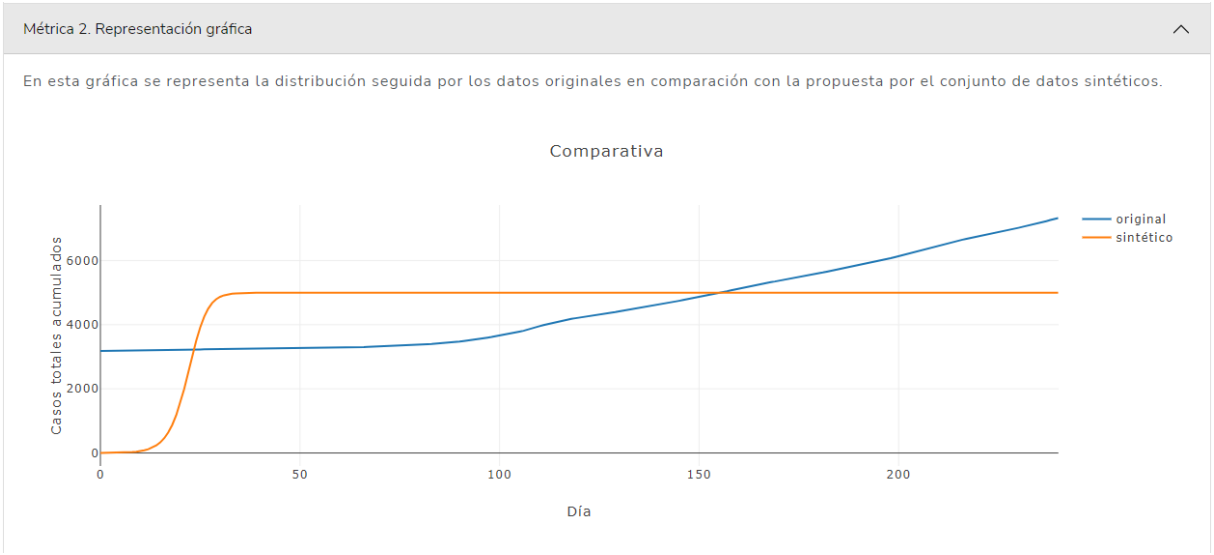


Figura 5.7: Ejemplo 2. Métrica 2

En las Figuras 5.8, 5.9 y 5.10 se puede observar la salida de la métrica 3. De las primeras dos imágenes se puede concluir que ambas colecciones tienen distinta simetría. Concretamente, la colección de datos originales presenta una distribución asimétrica con cola a la derecha (sesgada a la derecha), pues la media es mayor que la mediana o cuartil 2; y la colección de datos sintéticos presenta una distribución asimétrica con cola a la izquierda (sesgada a la izquierda), pues la media es menor que la mediana o cuartil 2. No obstante, estas imágenes también expresan que en la colección de datos sintéticos todos los valores son *outliers* (la barra azul en su totalidad sobrepasa el límite inferior rojo) mientras que en la original no es así.

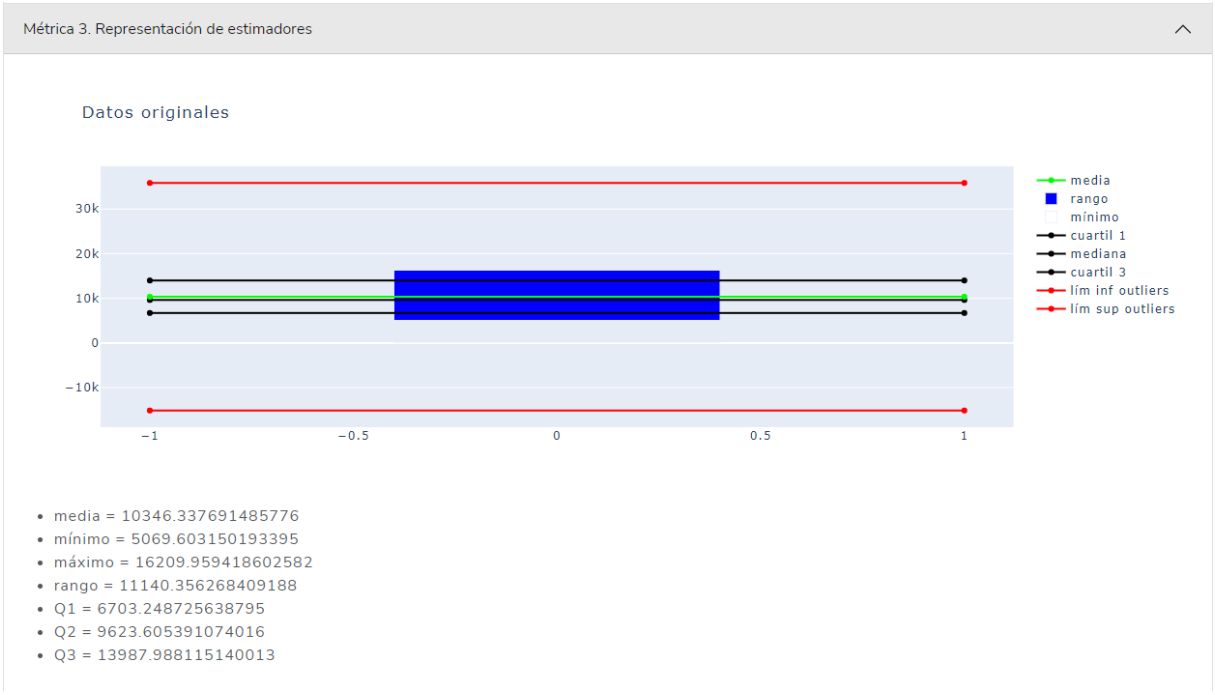


Figura 5.8: Ejemplo 2. Métrica 3.1

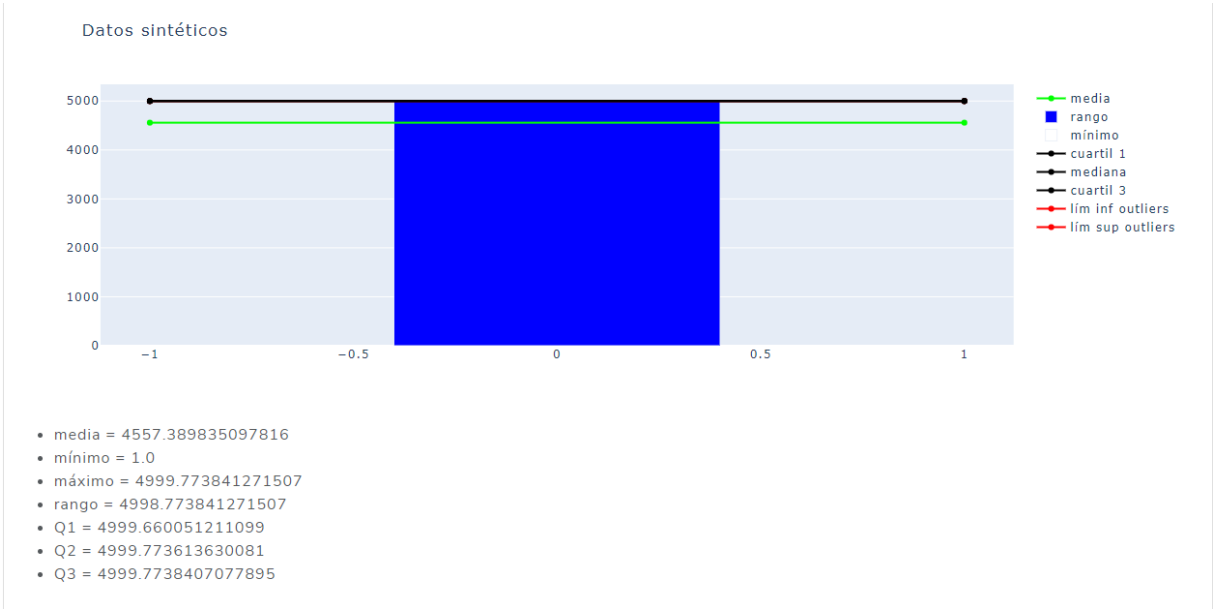


Figura 5.9: Ejemplo 2. Métrica 3.2

La presencia de *outliers* entre los datos sintéticos también se puede observar en el segundo diagrama de caja y bigotes de la Figura 5.10, el cual consta de una única línea, siendo, por tanto, todos los valores extremos. A pesar de que en estos diagramas no se representa la media, se observa que la mediana del conjunto de datos originales no está en el centro de la caja, de

donde se deduce que dicha muestra no es simétrica. Que la parte derecha de la caja sea mayor que la izquierda en el caso original, quiere decir que los casos totales acumulados originales comprendidos entre el 25 % y el 50 % de la población están menos dispersos que entre el 50 % y el 75 %. Esto puede observarse también en la concentración de los puntos representados debajo de dicha caja. El bigote de la izquierda es más corto que el de la derecha en el caso de los datos originales, por lo que el 25 % de los días con menos casos totales acumulados registrados están más concentrados que el 25 % de los días con más casos totales acumulados registrados. El rango intercuartílico, es decir, el 50 % de la población está comprendido en 2500 casos y 0 casos, para el conjunto de datos original y sintético, respectivamente [84].

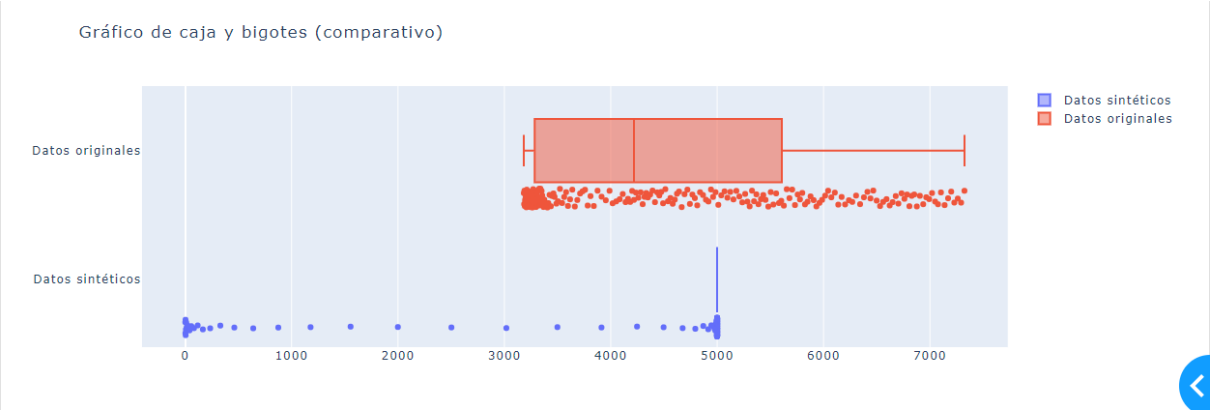


Figura 5.10: Ejemplo 2. Métrica 3.3

Ejemplo 3 A continuación, se listan los valores de cada uno de los parámetros, el caso escogido y se indica a qué país pertenece la comparación. Se puede observar que se trata de los mismos valores que en el ejemplo anterior excepto para los parámetros γ y σ, que invierten su relación, pasando a ser mayor γ que σ.

- Tamaño de la población: $N = 5000$
- Número de registros a generar menos uno: $L = 240$
- Tasa de recuperación: $\gamma = 0{,}5$
- Tasa de incubación: $\sigma = 0{,}1$
- Caso = 2 (Con medidas gubernamentales de contención)
- Tasa de infección antes de aplicar las medidas: $\beta_0 = 1$
- Día a partir del cual se comienza a aplicar las medidas (día $t_0 + 1$): $t_0 = 20$
- País = Reino Unido

En la Figura 5.11 se puede observar la salida de la métrica 1. Como comenta la última frase de la imagen, se obtiene una superposición "óptima" más cercana a 1 que a 0, por lo que la colección generada es válida en cuanto a utilidad retenida se refiere.

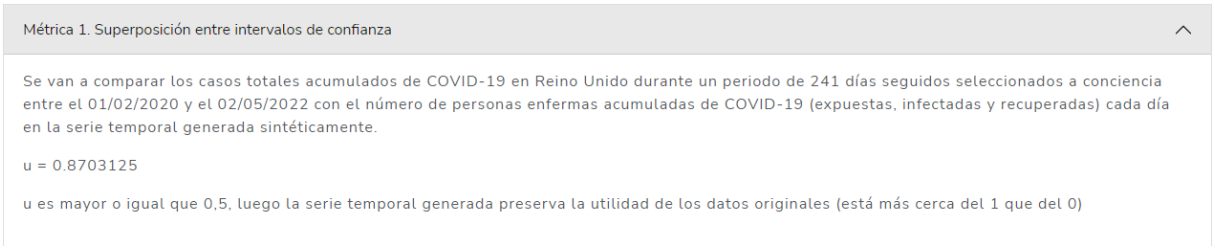


Figura 5.11: Ejemplo 3. Métrica 1

En la Figura 5.12 se puede observar la salida de la métrica 2. En esta gráfica se ve que ambas curvas son crecientes y suaves. La interpretación de las curvas resulta en que, en ambos casos, aproximadamente desde el día 80 hasta el 100 se produce una subida de casos totales acumulados. Las causas de que las curvas no sean coincidentes (aunque sí semejantes) se explican en el segundo párrafo de la subsección 5.2.

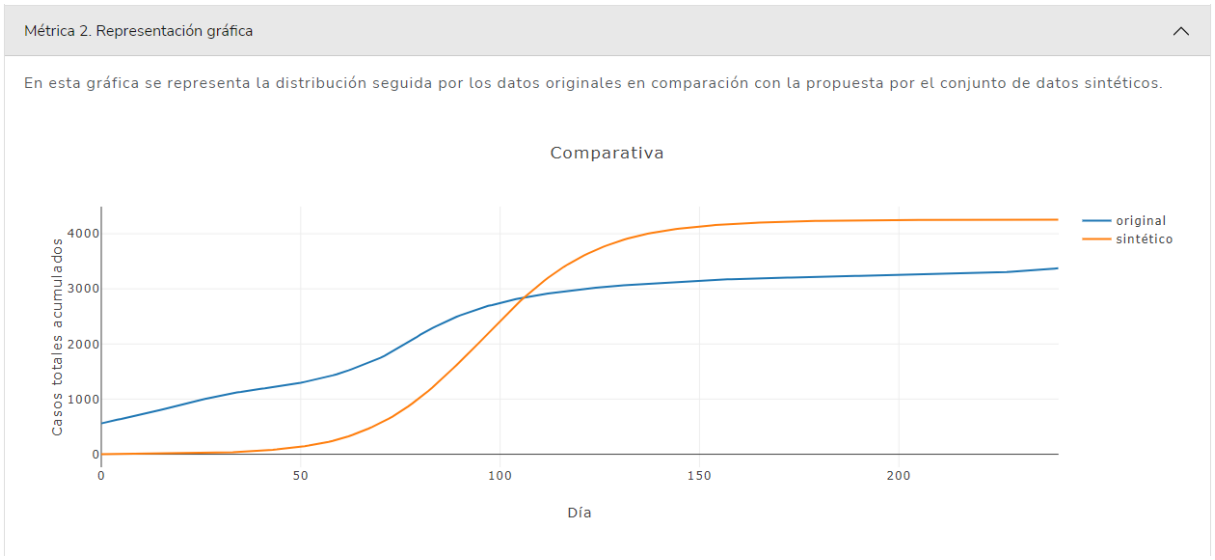


Figura 5.12: Ejemplo 3. Métrica 2

En las Figuras 5.13, 5.14 y 5.15 se puede observar la salida de la métrica 3. De las primeras dos imágenes se puede concluir que ambas colecciones tienen distinta simetría. Concretamente, la colección de datos originales presenta una distribución asimétrica con cola a la derecha (sesgada a la derecha), pues la media es mayor que la mediana o cuartil 2; y la colección de datos sintéticos presenta una distribución asimétrica con cola a la izquierda (sesgada a la izquierda), pues la media es menor que la mediana o cuartil 2. No obstante, estas imágenes también expresan que ninguna de las colecciones presenta *outliers* (la barra azul se encuentra entre las líneas rojas en ambos conjuntos).

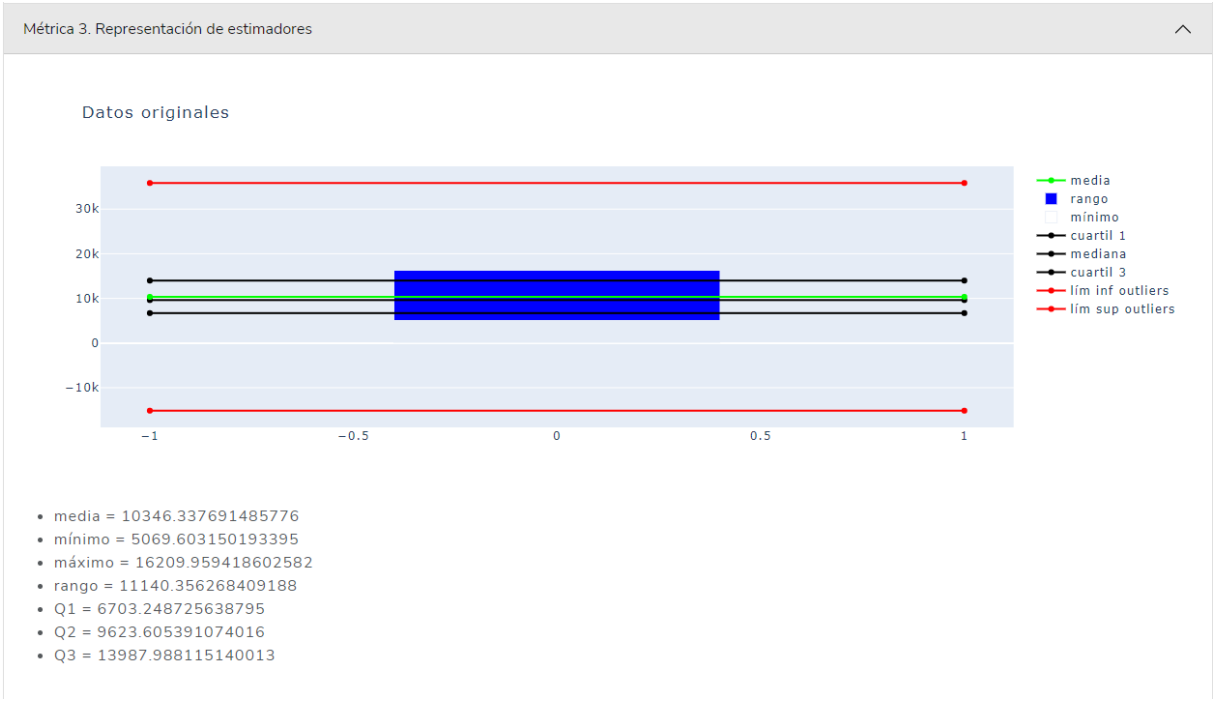


Figura 5.13: Ejemplo 3. Métrica 3.1

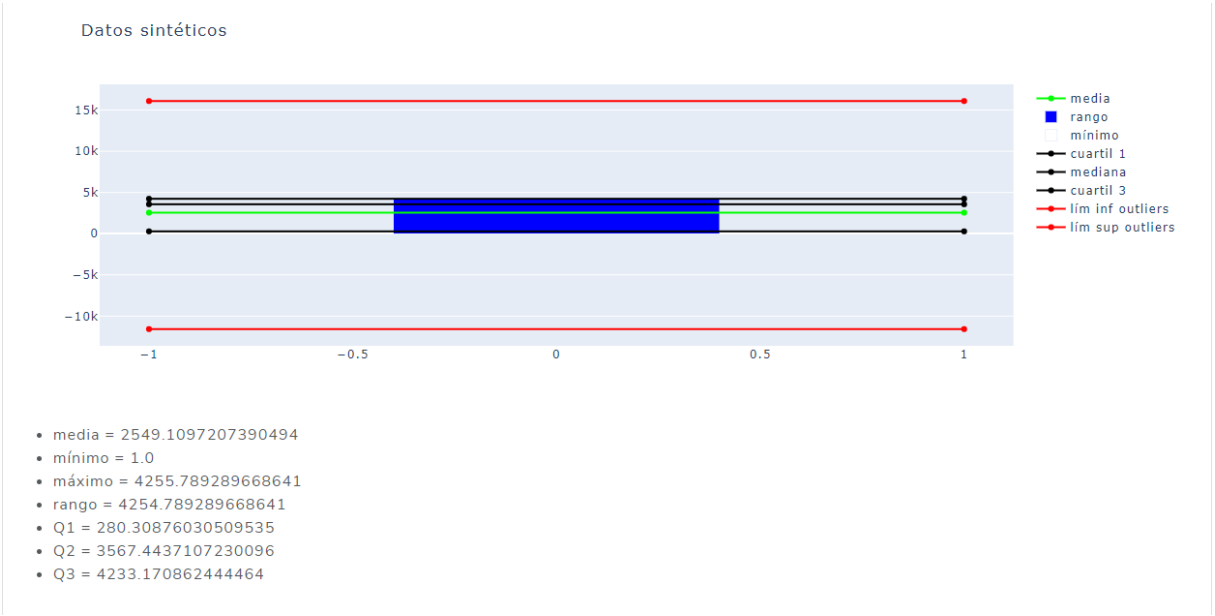


Figura 5.14: Ejemplo 3. Métrica 3.2

La Figura 5.15 corrobora la ausencia de *outliers* en ambos conjuntos de datos, ya que todos los puntos se sitúan entre los bigotes de cada gráfica en cada caso. A pesar de que en estos diagramas no se representa la media, se observa que la mediana de ambos conjuntos no está en el centro de la caja, de donde se deduce que ninguna de las muestras es simétrica. Que la parte izquierda

de la caja sea mayor que la de la derecha en ambos casos, quiere decir que los casos totales acumulados originales y sintéticos comprendidos entre el 25 % y el 50 % de la población están más dispersos que entre el 50 % y el 75 %. Esto puede observarse también en la concentración de los puntos representados debajo de cada caja. El bigote de la izquierda es más largo que el de la derecha en ambos casos, por lo que el 25 % de los días con más casos totales acumulados registrados están más concentrados que el 25 % de los días con menos casos totales acumulados registrados. El rango intercuartílico, es decir, el 50 % de la población está comprendido en 1800 casos y 4000 casos, para el conjunto de datos original y sintético, respectivamente [84].

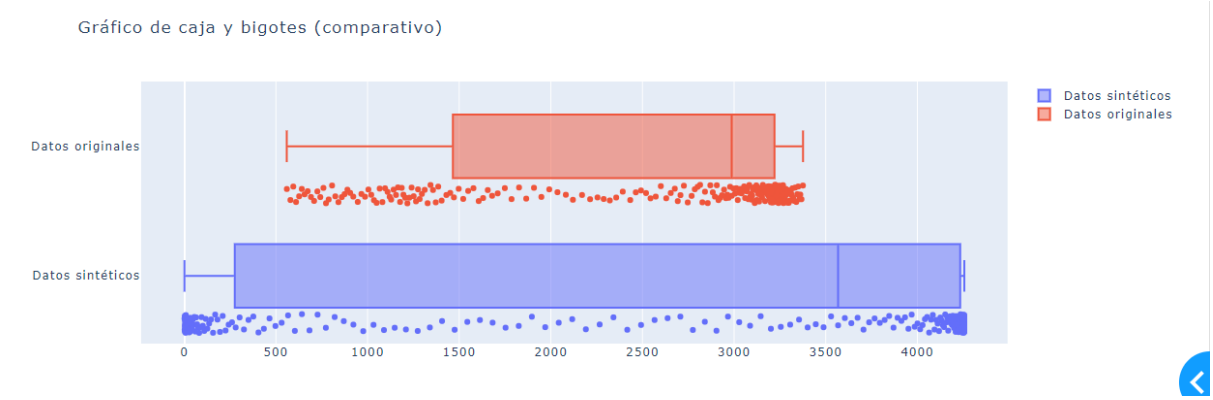


Figura 5.15: Ejemplo 3. Métrica 3.3

5.3. Discusión de resultados

En resumen, la medida u de utilidad basada en la superposición de los intervalos de confianza en la mayoría de las ocasiones sobrepasa el 0,5, por lo que se trata sin duda de datos que representan la realidad de forma fiable.

Por otro lado, ambas colecciones de datos tienen siempre la misma tendencia (curvas con la misma forma) pero difieren sus valores debido a, por ejemplo, sus condiciones iniciales diferentes. Se trata de curvas asimilables debido a que su forma es igual o se parece, pero no coincidentes. Estas diferencias se deben a lo que ya se comentó en la introducción (capítulo 1) de esta memoria y es que los datos originales están sesgados y no se ajustan de forma fidedigna a la realidad.

La presencia de *outliers* explica las diferencias existentes en estas evaluaciones. Sin embargo, no resulta sorprendente para un lector que haya vivido la pandemia, pues es bien conocido que los datos "reales" acerca del COVID-19 no han sido recogidos de forma continua a lo largo del tiempo, por lo que presentan un sesgo. Además, la pandemia ha sufrido distintas políticas de testeo, por lo que en muchas ocasiones el número de casos totales registrados no ha sido realmente el correcto ni consistente. Todas estas cuestiones y las ya comentadas con anterioridad no se pueden modelar, por lo que la distribución de la colección de datos sintéticos nunca va a ser idéntica a la de los datos reales.

Por último, cabe destacar que el modelo matemático SEIR utilizado no tiene en cuenta la posible reinfección de las personas recuperadas. Y, como se pone de manifiesto en los ejemplos 2 y 3, es necesario introducir valores lógicos de los parámetros para obtener resultados de evaluación favorable.

A pesar de todo, se puede observar en la sección 5.2 que los resultados obtenidos son positivos.

En resumen, los datos generados reflejan la realidad teórica, aunque no la de los datos recogidos por los países, de cuyo sesgo se han escrito muchas líneas durante todo el periodo de pandemia.

Capítulo 6

Conclusiones y trabajo futuro

De este capítulo se obtienen conclusiones a partir de la autocrítica (sección 6.1), y se depositan diversas líneas de trabajo que podrían retomarse en un futuro como continuación de este proyecto (sección 6.2).

6.1. Conclusiones

Esta sección tiene como finalidad que el autor de este proyecto reflexione acerca de lo que ha trabajado y si ha conseguido lo que deseaba. Para ello, consta de dos subsecciones bien diferenciadas entre sí; una pretende hacer retrospectiva sobre el proyecto (subsección 6.1.1) mientras que la otra se centra más en el estudiante (subsección 6.1.2).

6.1.1. Perspectiva del proyecto

Desde un punto de vista centrado en el proyecto, se van a tratar tres temas por separado.

- Objetivos específicos. Los objetivos marcados al principio del proyecto (sección 1.2) se cumplen satisfactoriamente. Por una parte, el objetivo 1.2, dividido en dos subobjetivos más específicos, se resuelve en su totalidad mediante el contenido bibliográfico del capítulo 3. Por otra parte, el sistema *software* desarrollado y explicado anteriormente en esta memoria satisface el objetivo 1.2, ya que, además de ser una herramienta de generación, a su vez contiene una parte que evalúa su propio trabajo (pestañas "Evaluación España" y "Evaluación Reino Unido"). Esta herramienta utiliza y aprovecha la investigación realizada para alcanzar el primer objetivo.
- Utilidad para el futuro. El sistema desarrollado en este proyecto puede parecer difícilmente escalable a otros asuntos totalmente ajenos a la pendemia debido a que este tema en el que se centra es cerrado. Sin embargo, esto no es así, pues basta con definir nuevas reglas que describan el comportamiento de un fenómeno para que esta herramienta genere datos sintéticos específicos para dicho fenómeno. Además, la separación entre las capas de la aplicación (*layout* y *callbacks*) y el uso de *Bootstrap* hacen de ésta una aplicación *web* fácilmente reutilizable. Además, existe quien piensa que no es útil desarrollar este tipo de herramientas que estudian un problema a posteriori, pero el hecho de que pueda ser reutilizado para otros problemas médicos o futuras nuevas pandemias lo convierte en un sistema de especial interés.

- Desarrollo del proyecto/metodología de trabajo. El uso de la metodología UVAGILE para el desarrollo del proyecto ha permitido que el resultado sea "óptimo", ya que sin la retroalimentación de cada *Sprint* proporcionada por los tutores y la comunidad no hubiera sido posible identificar los defectos del proyecto y revertirlos, en consecuencia, para mejorar su calidad.

6.1.2. Perspectiva personal

Desde un punto de vista personal, el desarrollo de este proyecto me ha hecho crecer como estudiante y como futura ingeniera informática en ciertos aspectos que se comentan a continuación. El hecho de elegir el tema de este TFG me ha permitido perder el miedo a la toma de decisiones importantes. Además, el proceso de esta elección me llevó a descartar otros temas y definir mis gustos dentro de esta ingeniería. En relación con esto, he descubierto lo interesantes que son los datos sintéticos y no descarto seguir investigando acerca de ellos. Por otra parte, nunca antes me había enfrentado a la realización de un proyecto de tal envergadura y comprender los formalismos que conlleva me ha permitido observar el tipo de trabajos a los que debo enfrentarme tras finalizar el grado.

Además, como alumna del programa de estudios conjunto en Matemáticas e Ingeniería Informática de Servicios y Aplicaciones, el hecho de poder relacionar ambos TFGs, el de Matemáticas y el de Ingeniería Informática, me ha permitido cerrar esta etapa universitaria con la visión global de ambas ciencias que necesitaba desde que comencé mis estudios en la Universidad de Valladolid. Tras este proyecto, he podido comprender realmente la importancia de las matemáticas y la ingeniería informática como conjunto, es decir, el solapamiento y la ayuda en ambos sentidos que la ciencia y la ingeniería se ofrecen. Concretamente, la construcción de este sistema *software* no hubiera sido posible sin la base matemática en la que está basado el generador, el modelo SEIR que describe la pandemia. Por otro lado, el modelo SEIR, y en general cualquier modelización matemática, no puede ser aprovechado sin las herramientas que ofrece la ingeniería, como en este caso las funciones de Python que permiten resolver el sistema de ecuaciones diferenciales ordinarias que comprende el modelo. Por si fuera poco, mi TFG de Matemáticas, titulado "Modelización de redes neuronales aplicadas a la evolución y seguimiento del Covid-19", consta de un modelo de ML que hay que entrenar con datos acerca del COVID-19, por lo que se podría abastecer de los datos generados por el sistema construido en mi TFG de Informática. Creo que esta es la mejor forma de cerrar el estudio de ambos grados.

Ahora bien, de forma más concreta, me he enfrentado a ciertos bloqueos que he conseguido superar; este es el mayor aprendizaje que obtengo de este trabajo. En primer lugar, fue desafiante la elección de una técnica de generación que aunara sencillez, utilidad, privacidad y rapidez. Tras estudiar el estado del arte, reparé en cuál sería la más adecuada pero aun así no me veía capaz de encontrar el modo de usar dicha técnica. Después, me resultó complicado el proceso de análisis y diseño *software* debido a que no sabía qué iba a ser capaz de construir con *Dash*, *framework* completamente nuevo para mí. Esta incertidumbre no me permitía visualizar la futura herramienta con claridad. Con la ayuda de mis tutores y algo de imaginación hice un prototipo de mi sistema. El siguiente reto al que me enfrenté llegó con la implementación del primer caso de uso; no sabía de qué forma introducir la aleatoriedad en el Caso 2. A base de pruebas conseguí llegar a la función definida a trozos que consolida el modelo SEIR actualmente. La mayor dificultad vino de la mano de la implementación del segundo caso de uso; debía guardar la colección generada para poder después representarla y es en este momento cuando tuve que

contemplar la definición de una clase con sus atributos. Tuve que informarme acerca de los *dataframes* de Pandas para la representación tabular, y es un conocimiento fundamental que me alegra haber adquirido. Después de varias reyertas con los *callbacks*, la funcionalidad estaba aproximadamente preparada. Solo faltaba dar una buena apariencia a la aplicación modificando su *layout*. Para ello, se me ocurrió que lo mejor sería usar un tema de *Bootstrap*. Este paso, aunque sencillo, me permitió también ampliar mi conocimiento sobre el uso de esta biblioteca y el diseño *web*.

6.2. Trabajo futuro

De cara a una continuación futura del proyecto, se pueden cambiar las reglas de acuerdo a otros modelos matemáticos que tienen en cuenta más detalles sobre el desarrollo de la pandemia y que permiten, por tanto, obtener resultados más utiles que los obtenidos con el sistema actual. Los propios modelos SEIR pueden ser modificados para tener en cuenta más supuestos según las características de un virus específico.

En [85] se proporciona un modelo SEIR que difiere del explicado en la sección 4.1 en que involucra aspectos demográficos, esto es, la población total $N(t)$ evoluciona a lo largo del tiempo t. De este modo, el sistema de ecuaciones diferenciales ordinarias formado por las Ecuaciones 4.1, 4.2, 4.3 y 4.4 se modifica de acuerdo a esto como se observa en las Ecuaciones 6.1, 6.2, 6.3 y 6.4.

$$S'(t) = -\beta \cdot S(t) \cdot I(t) \tag{6.1}$$
$$E'(t) = \beta \cdot S(t) \cdot I(t) - \sigma \cdot E(t) \tag{6.2}$$
$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) \tag{6.3}$$
$$R'(t) = \gamma \cdot I(t) \tag{6.4}$$

La **tasa de natalidad** ν de la población también puede ser considerada. Se asume que las personas nacen sanas, y se agrega un término $\nu \cdot N(t)$ a la primera ecuación. De este modo, el sistema de ecuaciones diferenciales ordinarias formado por las Ecuaciones6.1, 6.2, 6.3 y 6.4 se modifica de acuerdo a esto como se observa en las Ecuaciones 6.5, 6.6, 6.7 y 6.8.

$$S'(t) = -\beta \cdot S(t) \cdot I(t) + \nu \cdot N(t) \tag{6.5}$$
$$E'(t) = \beta \cdot S(t) \cdot I(t) - \sigma \cdot E(t) \tag{6.6}$$
$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) \tag{6.7}$$
$$R'(t) = \gamma \cdot I(t) \tag{6.8}$$

De igual manera, el esquema de la Figura 4.1 se modifica de acuerdo a esto como se observa en la Figura 6.1.

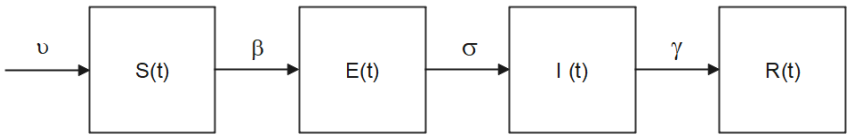


Figura 6.1: "Diagrama de flujo" del modelo SEIR (Variante: tasa de natalidad)

Finalmente, se completa añadiendo la **tasa de mortalidad** μ de la población. Se considera que una persona puede morir sin importar su estado (S,E,I o R), y de causa no necesariamente ligada a la epidemia. Se retiran entonces estas personas de cada ecuación (esto es, $-\mu \cdot S(t)$, $-\mu \cdot E(t)$, $-\mu \cdot I(t)$ o $-\mu \cdot R(t)$, según la subpoblación considerada). De este modo, el sistema de ecuaciones diferenciales ordinarias formado por las Ecuaciones 6.5, 6.6, 6.7 y 6.8 se modifica de acuerdo a esto como se observa en las Ecuaciones 6.9, 6.10, 6.11 y 6.12.

$$S'(t) = -\beta \cdot S(t) \cdot I(t) + \nu \cdot N(t) - \mu \cdot S(t) \tag{6.9}$$
$$E'(t) = \beta \cdot S(t) \cdot I(t) - \sigma \cdot E(t) - \mu \cdot E(t) \tag{6.10}$$
$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) - \mu \cdot I(t) \tag{6.11}$$
$$R'(t) = \gamma \cdot I(t) - \mu \cdot R(t) \tag{6.12}$$

De igual manera, el esquema de la Figura 6.1 se modifica de acuerdo a esto como se observa en la Figura 6.2.

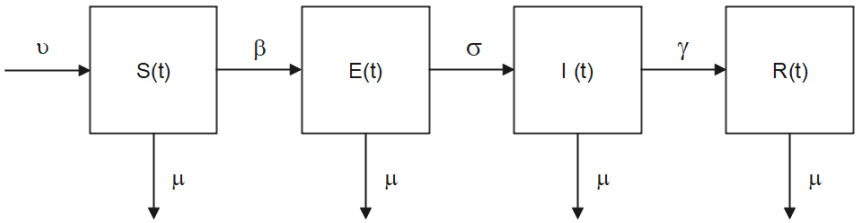


Figura 6.2: "Diagrama de flujo" del modelo SEIR (Variante: tasa de mortalidad)

Para ciertos virus, es necesario considerar la **pérdida de inmunidad**, es decir, considerar que existen personas recuperadas que vuelven a ser sanas, por lo que son susceptibles (con una tasa δ) de ser infectadas nuevamente. De este modo, el sistema de ecuaciones diferenciales ordinarias formado por las Ecuaciones 6.9, 6.10, 6.11 y 6.12 se modifica de acuerdo a esto como se observa en las Ecuaciones 6.13, 6.14, 6.15 y 6.16.

$$S'(t) = -\beta \cdot S(t) \cdot I(t) + \nu \cdot N(t) - \mu \cdot S(t) + \delta \cdot R(t) \tag{6.13}$$
$$E'(t) = \beta \cdot S(t) \cdot I(t) - \sigma \cdot E(t) - \mu \cdot E(t) \tag{6.14}$$
$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) - \mu \cdot I(t) \tag{6.15}$$
$$R'(t) = \gamma \cdot I(t) - \mu \cdot R(t) - \delta \cdot R(t) \tag{6.16}$$

De igual manera, el esquema de la Figura 6.2 se modifica de acuerdo a esto como se observa en la Figura 6.3.

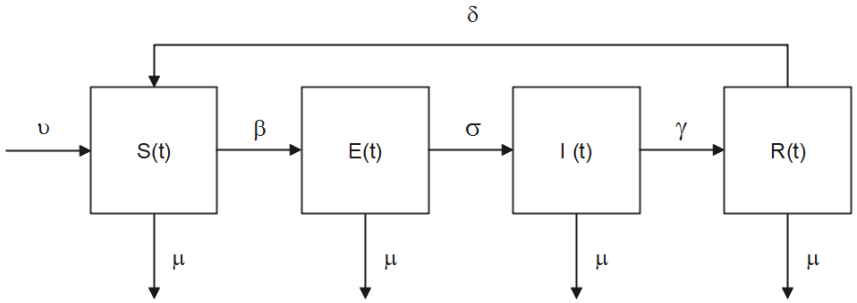


Figura 6.3: "Diagrama de flujo" del modelo SEIR (Variante: pérdida de inmunidad)

Para evitar o limitar las consecuencias de una pérdida de inmunidad, una política de vacunación (con una **tasa de vacunación** ϵ) puede ser puesta en práctica. En este caso, las personas sanas pasan a ser, por tanto, directamente recuperadas. De este modo, el sistema de ecuaciones diferenciales ordinarias formado por las Ecuaciones 6.9, 6.10, 6.11 y 6.12 se modifica de acuerdo a esto como se observa en las Ecuaciones 6.17, 6.18, 6.19 y 6.20.

$$S'(t) = -\beta \cdot S(t) \cdot I(t) + \nu \cdot N(t) - \mu \cdot S(t) - \epsilon \cdot S(t) \tag{6.17}$$
$$E'(t) = \beta \cdot S(t) \cdot I(t) - \sigma \cdot E(t) - \mu \cdot E(t) \tag{6.18}$$
$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) - \mu \cdot I(t) \tag{6.19}$$
$$R'(t) = \gamma \cdot I(t) - \mu \cdot R(t) + \epsilon \cdot S(t) \tag{6.20}$$

De igual manera, el esquema de la Figura 6.2 se modifica de acuerdo a esto como se observa en la Figura 6.4.

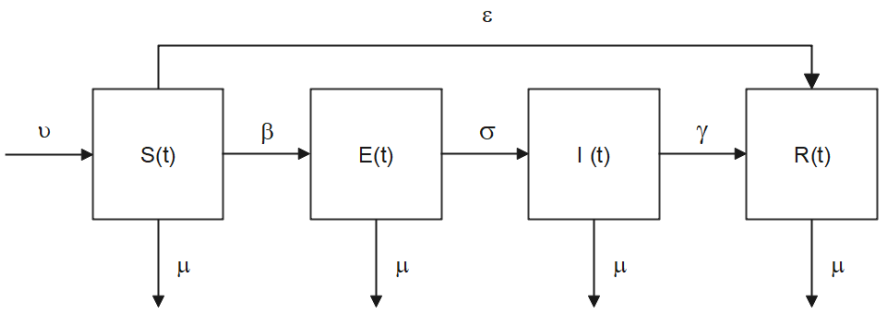


Figura 6.4: "Diagrama de flujo" del modelo SEIR (Variante: tasa de vacunación)

Algunas madres infectadas pueden transmitir directamente el virus al **recién nacido** (que, potencialmente, es directamente **infectado**). De este modo, el sistema de ecuaciones diferenciales ordinarias formado por las Ecuaciones 6.17, 6.18, 6.19 y 6.20 se modifica de acuerdo a esto como se observa en las Ecuaciones 6.21, 6.22, 6.23 y 6.24.

$$S'(t) = -\beta \cdot S(t) \cdot I(t) + \nu \cdot N(t) - \mu \cdot S(t) - \epsilon \cdot S(t) \tag{6.21}$$
$$E'(t) = \beta \cdot S(t) \cdot I(t) - \sigma \cdot E(t) - \mu \cdot E(t) \tag{6.22}$$
$$I'(t) = \sigma \cdot E(t) - \gamma \cdot I(t) - \mu \cdot I(t) - \nu \cdot I(t) \tag{6.23}$$
$$R'(t) = \gamma \cdot I(t) - \mu \cdot R(t) + \epsilon \cdot S(t) \tag{6.24}$$

De igual manera, el esquema de la Figura 6.4 se modifica de acuerdo a esto como se observa en la Figura 6.5.

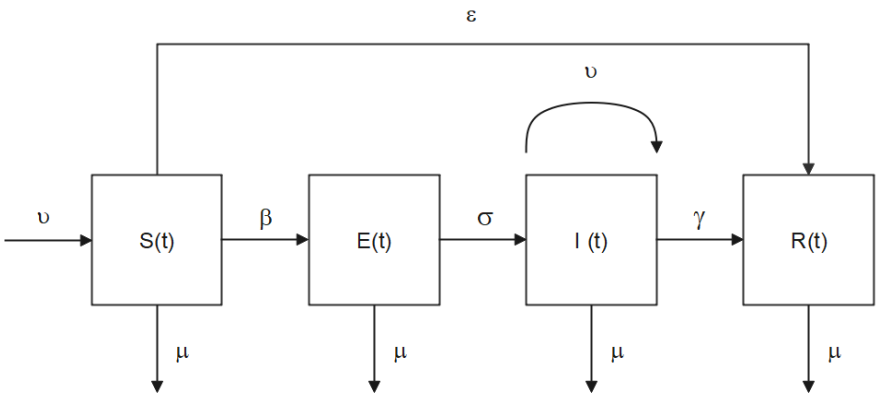


Figura 6.5: "Diagrama de flujo" del modelo SEIR (Variante: recién nacido infectado)

Otra mejora del proyecto que puede considerarse en un futuro es potenciar el *layout* de la *app*. Por ejemplo, se puede pulir la tabla, añadiendo funcionalidades como la capacidad de editar filas y/o columnas. En definitiva, se puede conseguir desarrollar una tabla más dinámica. Por último, se puede ampliar la evaluación de la colección usando técnicas estadísticas más complejas que permitan obtener resultados más consistentes que los actuales.

Parte I

Apéndices

Apéndice A

Acrónimos

LOPD Ley Orgánica de Protección de Datos

GDPR Reglamento General de Protección de Datos o *General Data Protection Regulation*

RTVE Radio TV Española

OMS Organización Mundial de la Salud

UCI Unidad de Cuidados Intensivos

PCR Reacción en Cadena de la Polimerasa o *Polymerase Chain Reaction*

RRHH Recursos Humanos

TFG Trabajo de Fin de Grado

ECTS Sistema Europeo de Transferencia y Acumulación de Créditos o *European Credit Transfer System*

PLN Procesamiento del Lenguaje Natural o *Natural Language Processing*

NLU Comprensión del Lenguaje Natural o *Natural Language Understanding*

IA Inteligencia Artificial o (*Artificial Intelligence*)

ML Aprendizaje Automático o *Machine Learning*

DL Aprendizaje Profundo o *Deep Learning*

NN Red Neuronal o *Neural Network*

ANN Red Neuronal Artificial o *Artificial Neural Network*

SNN Red Neuronal Simulada o *Simulated Neural Network*

SVM Máquina de Vectores de Soporte o *Support-Vector Machine*

VAE Autocodificador Variacional o *Variational Autoencoder*

GAN Red Generativa Adversaria o *Generative Adversarial Network*

IM Imputación Múltiple

CRM Gestión de Relación con los Clientes o *Customer Relationship Management*

PII Información de Identificación Personal o *Personally Identifiable Information*

IU Interfaz de Usuario o *User Interface*

IDE Entorno de Desarrollo Integrado o *Integrated Development Environment*

BD Base de Datos o *Database*

CSV Valores Separados por Comas o *Comma Separated Values*

URL Localizador Uniforme de Recursos o *Uniform Resource Locator*

Bibliografía

[1] CIO Applications Europe. *Data-Driven Business Environment: A Perfect Decision-Making Solution*. URL: https://www.cioapplicationseurope.com/news/datadriven-business-environment-a-perfect-decisionmaking-solution--nid-1324.html?msclkid=b53e4450a6ac11ecbbc80f6ddc1e16b0 (visitado 14-06-2022).

[2] Gobierno de España. *BOE.es - BOE-A-2018-16673 Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.* 2018. URL: https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673 (visitado 06-07-2022).

[3] Elise Devaux. «Types of synthetic data and 5 real-life examples - Statice». En: (2021), págs. 1-10. URL: https://www.statice.ai/post/types-synthetic-data-examples-real-life-examples.

[4] Business Insider. *Business Insider España: Actualidad económica, tendencias y mundo global*. URL: https://www.businessinsider.es/ (visitado 11-09-2022).

[5] Alberto R. Aguiar. *Qué son los datos sintéticos y por qué empresas querrán protegerlos | Business Insider España*. URL: https://www.businessinsider.es/son-datos-sinteticos-empresas-querran-protegerlos-1013387 (visitado 11-09-2022).

[6] T.M. Mitchell. *Machine Learning*. 1997, Caps 3, 6, 8 y 10.

[7] Wikipedia. *Synthetic data - Wikipedia*. URL: https://en.wikipedia.org/wiki/Synthetic_data?msclkid=1bd16a5ab44a11ecbf7292cf7f5236d3#History (visitado 11-06-2022).

[8] Donald Rubin. «Discussion: Statistical Disclosure Limitation». En: *Journal of Official Statistics* 9.Discussion: Statistical Disclosure Limitation (1993), págs. 461-468.

[9] John M. Abowd. «Confidentiality Protection of Social Science Micro Data: Synthetic Data and Related Methods. [Powerpoint slides]». En: (2011).

[10] Roderick J.A. Little. «Statistical Analysis of Masked Data». En: *Journal of Official Statistics* 9 (1993), págs. 407-426.

[11] Janet Slifka. *Tools for Generating Synthetic Data Helped Bootstrap Alexa's New-Language Releases*. 2019. URL: https://www.amazon.science/blog/tools-for-generating-synthetic-data-helped-bootstrap-alexas-new-language-releases (visitado 07-09-2022).

[12] Kyle Wiggers. *The challenges of developing autonomous vehicles during a pandemic*. 2020. URL: https://venturebeat.com/ai/challenges-of-developing-autonomous-vehicles-during-coronavirus-covid-19-pandemic/ (visitado 07-09-2022).

[13] Jonathan Vanian. *American Express tests deepfake technology to fight financial fraud | Fortune*. URL: https://fortune.com/2020/09/03/american-express-deepfake-artificial-intelligence/ (visitado 07-09-2022).

[14] Elise Devaux. *Privacy-preserving machine learning in insurance: La Mobilière success story - Statice*. 2020. URL: https://www.statice.ai/post/future-proofing-data-operations-successful-insurance-mobiliere (visitado 07-09-2022).

[15] Elise Devaux. *Testing synthetical clinical data for innovation in healthcare with Roche - Statice*. URL: https://www.statice.ai/post/data-innovation-healthcare-synthetical-clinical-data-roche (visitado 07-09-2022).

[16] UNESCO. *¿Cómo la inteligencia artificial puede ayudar a combatir el COVID-19?* 2020. URL: https://www.eltambor.es/como-puede-la-inteligencia-artificial-ayudar-a-combatir-pandemias-como-el-covid-19/ (visitado 20-08-2022).

[17] Beatriz Asuar. *Fallecidos por coronavirus: ¿Qué ocurre con los datos de Sanidad y por qué pueden darse más desajustes? | Público*. URL: https://www.publico.es/sociedad/fallecidos-coronavirus-ocurre-datos-sanidad-darse-desajustes.html#analytics-noticia:contenido-enlace (visitado 11-09-2022).

[18] Antena 3 Noticias. *Coronavirus: ¿Podría haber fallado Excel en el recuento de datos de nuevos contagios en Reino Unido?* URL: https://www.antena3.com/noticias/mundo/podria-haber-fallado-excel-en-el-recuento-de-datos-de-nuevos-contagios-en-reino-unido_202010065f7c113a3d763700016709f4.html (visitado 11-09-2022).

[19] Santiago Fernandéz. «Series Temporales Introducción». En: (2017), pág. 2. URL: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/EDescrip/tema7.pdf.

[20] Miguel A. Martínez-Prieto y col. «Agilizando el aprendizaje de bases de datos». En: *Actas de las JENUI* 6 (2021), págs. 83-90.

[21] Joel Francia. *¿Qué es Scrum? | Scrum.org*. 2017. URL: https://www.scrum.org/resources/blog/que-es-scrum (visitado 13-09-2022).

[22] Derek Davidson. *¿Por qué usamos Story Points para estimar? | Scrum.org*. URL: https://www.scrum.org/resources/blog/why-do-we-use-story-points-estimating (visitado 13-06-2022).

[23] Trello. *Manage Your Team's Projects From Anywhere | Trello*. URL: https://trello.com/ (visitado 11-07-2022).

[24] Glassdoor. *Sueldos de la empresa | Glassdoor.es*. 2021. URL: https://www.glassdoor.es/Sueldos/index.htm (visitado 07-06-2022).

[25] LinkedIn. *LinkedIn Salary: Descubre sueldos reales. Conoce hasta dónde puedes llegar | LinkedIn*. URL: https://www.linkedin.com/salary/ (visitado 11-07-2022).

[26] Javier Santos Pascualena. *¿Cuánto cuesta contratar un trabajador? - Infoautonomos*. 2021. URL: https://www.infoautonomos.com/blog/cuanto-cuesta-contratar-un-trabajador/ (visitado 16-06-2022).

[27] The Adecco Group Institute. *Qué Es Data Driven Y Cómo Ayuda A Transformar Los Negocios - Adecco Institute*. URL: https://www.adeccoinstitute.es/articulos/que-es-data-driven/ (visitado 14-06-2022).

[28] DELL Technologies. *Global Data Protection Index Report | Dell Technologies Singapore*. URL: https://www.dell.com/en-sg/dt/data-protection/gdpi/index.htm (visitado 14-06-2022).

[29] Latam Inversor. *Las organizaciones administran casi un 40 % más de datos que hace un año | Inversor Latam*. URL: https://inversorlatam.com/las-organizaciones-administran-casi-un-40-mas-de-datos-que-hace-un-ano/ (visitado 15-06-2022).

[30] FutureCIO Editors. *Part 1: The importance of data & AI/ML in data analysis - FutureCIO*. URL: https://futurecio.tech/the-importance-of-data-ai-ml-in-data-analysis/ (visitado 14-06-2022).

[31] Anders Norén. *5 maneras de lidiar con la falta de datos en el aprendizaje automático*. 2019. URL: https://sitiobigdata.com/2019/12/24/5-maneras-de-lidiar-con-la-falta-de-datos-en-el-aprendizaje-automatico/ (visitado 14-06-2022).

[32] Cem Dilmegani. *in-Depth Synthetic Data Guide: What is it?How does it enable AI?* URL: https://research.aimultiple.com/synthetic-data/ (visitado 14-06-2022).

[33] UNIR. *Big data y Machine Learning: ¿cómo se complementan?* 2020. URL: https://www.unir.net/ingenieria/revista/big-data-machine-learning/ (visitado 14-06-2022).

[34] Andrew Entwistle. «WHAT IS ARTIFICIAL INTELLIGENCE?» En: *Engineering materials and design* 32.3 (1988). ISSN: 00138045. DOI: 10.7551/mitpress/12518.003.0004. URL: http://www-formal.stanford.edu/jmc/.

[35] Ana Casali. «¿ Qué es la Inteligencia Artificial ? ¿ Qué es la Inteligencia Artificial ?» En: *Researchgate* 1-6.March (2015). URL: https://www.researchgate.net/publication/268275299_Que_es_la_Inteligencia_Artificial.

[36] *¿Qué es la inteligencia artificial (IA)? - España*. IBM. Agosto 2015. URL: https://www.ibm.com/es-es/cloud/learn/what-is-artificial-intelligence.

[37] Anibal Bregón. «Tema 7 - Aprendizaje - Introducción». En: (2021).

[38] M A Gutiérrez Naranjo F J Martín Mateos J L Ruiz Reina. «Tema 6: Introducción al aprendizaje automático». En: (2021).

[39] Agenciab12. *Etapas del proceso de Machine Learning - Agencia B12*. 2020. URL: https://agenciab12.com/noticia/etapas-proceso-machine-learning (visitado 10-07-2022).

[40] IBM Cloud Education. *¿Qué son las redes neuronales? - España | IBM*. 2020. URL: https://www.ibm.com/es-es/cloud/learn/neural-networks (visitado 10-07-2022).

[41] F J Martín Mateos J L Ruiz Reina. «Tema 7: Introducción a las redes neuronales». En: (2021).

[42] Anibal Bregón. «Tema 11 - Aprendizaje - DL». En: (2021).

[43] Mat Powell. *¿Qué es el aprendizaje social?* 2021. URL: https://www.netapp.com/es/artificial-intelligence/what-is-deep-learning/ (visitado 11-06-2022).

[44] Li Yang y Abdallah Shami. «On hyperparameter optimization of machine learning algorithms: Theory and practice». En: *Neurocomputing* 415 (nov. de 2020), págs. 295-316. ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2020.07.061. arXiv: 2007.15745.

[45] Juan Ignacio Bagnato. *Ejemplo Regresión Lineal Python | Aprende Machine Learning*. 2018. URL: https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/ (visitado 15-06-2022).

[46] IBM. *Acerca de la regresión lineal | IBM*. URL: https://www.ibm.com/es-es/topics/linear-regression?mhsrc=ibmsearch_a&mhq=REGRESIONlineal (visitado 11-09-2022).

[47] IBM. *¿Qué es un árbol de decisión? | IBM.* URL: https://www.ibm.com/es-es/topics/decision-trees (visitado 11-09-2022).

[48] Lucidchart; *What is a Decision Tree Diagram.* 2020. URL: https://www.ibm.com/topics/decision-treeshttps://www.lucidchart.com/pages/decision-tree (visitado 15-06-2022).

[49] IBM Cloud Education. *What is Boosting? | IBM.* URL: https://www.ibm.com/cloud/learn/boosting (visitado 21-06-2022).

[50] Edward Beltrami. *What Is Random?* 2020. DOI: 10.1007/978-1-0716-0799-2. URL: https://www.ibm.com/cloud/learn/random-forest.

[51] IBM. *Modelos de máquina de vectores de soporte - Documentación de IBM.* URL: https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=nodes-support-vector-machine-models (visitado 15-06-2022).

[52] Joseph Rocca. «Understanding Variational Autoencoders (VAEs) | by Joseph Rocca | Towards Data Science». En: *Towards Data Science* (2019), págs. 1-24. URL: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[53] Gem Dilmegani. *Synthetic Data Generation: Techniques, Best Practices & Tools.* 2021. URL: https://research.aimultiple.com/synthetic-data-generation/ (visitado 14-06-2022).

[54] European Union. *What is GDPR, the EU's new data protection law? - GDPR.eu.* 2018. URL: https://gdpr.eu/what-is-gdpr/ (visitado 13-09-2022).

[55] Ashish Dandekar, Remmy A M Zen y Stephane Bressan. «Comparative Evaluation of Synthetic Dataset Generation Methods». En: *Proceedings of ACM Conference (Deep Learning Security Workshop).* (2019).

[56] Kajal Singh. *Datos sintéticos: beneficios clave, tipos, métodos de generación y desafíos | por Kajal Singh | Hacia la ciencia de datos.* URL: https://towardsdatascience.com/synthetic-data-key-benefits-types-generation-methods-and-challenges-11b0ad304b55 (visitado 15-09-2022).

[57] Syntho. *¿Qué son los datos sintéticos y en qué se diferencian los datos sintéticos generados por IA?* URL: https://syntho.ai/es/what-is-synthetic-data/ (visitado 14-06-2022).

[58] Manuel Pasieka. *The evolution of synthetic data: a comparison of three data generation methods - MOSTLY AI.* URL: https://mostly.ai/2020/10/28/comparison-of-synthetic-data-types/ (visitado 15-06-2022).

[59] Cem Dilmegani. *4 Synthetic data applications to enable finance innovation in '22.* 2021. URL: https://research.aimultiple.com/synthetic-data-finance/ (visitado 15-09-2022).

[60] Joanna Kamińska. *How to use synthetic data in Machine Learning and AI - Statice.* URL: https://www.statice.ai/post/synthetic-data-machine-learning (visitado 15-09-2022).

[61] Miguel Platzer. *Aumente la precisión de su aprendizaje automático con datos sintéticos, principalmente IA.* URL: https://mostly.ai/blog/boost-machine-learning-accuracy-with-synthetic-data/ (visitado 14-09-2022).

[62] Ashish Dandekar, Remmy A M Zen y Stéphane Bressan. «Introduction Synthetic Data Generation Data Synthesizers Experiments Conclusion References Comparative Evaluation of Synthetic Dataset Generation Methods». En: (2017).

[63] Daren S. Yates, Daniel S.; Moore, David S; Starnes. *The Practice of Statistics*. Ed. por W. H. Freeman and Company. 2.ª ed. New York, 2003. ISBN: 978-0-7167-4773-4.

[64] Enciclopedia Económica. *Variable cualitativa - ¿Qué es?, características, ejemplos y más*. 2019. URL: https://enciclopediaeconomica.com/variable-cualitativa/ (visitado 15-09-2022).

[65] Luis Benites. *Imputación múltiple para datos faltantes: definición, descripción general en 2022 → STATOLOGOS®*. URL: https://statologos.com/imputacion-multiple/ (visitado 16-06-2022).

[66] Tirtha Sarkar. *Statistical Modeling with Python: How-to & Top Libraries - Kite Blog*. URL: https://www.kite.com/blog/python/statistical-modeling-python-libraries/?msclkid=07c113aaa88a11ec8e4fd897bd84f05b (visitado 15-06-2022).

[67] Christoph Wehmeyer. *How do you generate synthetic data? - Statice*. URL: https://www.statice.ai/post/how-generate-synthetic-data (visitado 15-06-2022).

[68] Python Software Foundation. *General Python FAQ — Python 3.10.4 documentation*. 2022. URL: https://docs.python.org/3/faq/general.html#what-is-python (visitado 15-06-2022).

[69] José Manuel Gutiérrez y Juan Luis Varona. «Análisis de la posible evolución de la epidemia de coronavirus COVID-19 por medio de un modelo SEIR». En: *Departamento de Matemáticas y Computación* (2020), págs. 1-14.

[70] Ana Pais. *Modelos matemáticos de coronavirus: por qué el más popular para predecir la curva del covid-19 considera a los muertos como recuperados BBC News Mundo*. 2020. URL: https://www.bbc.com/mundo/noticias-52455414 (visitado 15-09-2022).

[71] Qianying Lin y col. «A conceptual model for the coronavirus disease 2019 (COVID-19) outbreak in Wuhan, China with individual reaction and governmental action». En: *International Journal of Infectious Diseases* 93 (abr. de 2020), págs. 211-216. ISSN: 1201-9712. DOI: 10.1016/J.IJID.2020.02.058. URL: http://www.ijidonline.com/article/S120197122030117X/fulltext.

[72] Joseph T. Wu, Kathy Leung y Gabriel M. Leung. «Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study». En: *The Lancet* 395.10225 (feb. de 2020), págs. 689-697. ISSN: 1474547X. DOI: 10.1016/S0140-6736(20)30260-9/ATTACHMENT/6C640CCE-3F4E-4101-92DC-1AE5C267F01A/MMC1.PDF. URL: http://www.thelancet.com/article/S0140673620302609/fulltext.

[73] Biao Tang y col. «An updated estimation of the risk of transmission of the novel coronavirus (2019-nCov)». En: *Infectious Disease Modelling* 5 (ene. de 2020), págs. 248-255. ISSN: 2468-0427. DOI: 10.1016/J.IDM.2020.02.001.

[74] Inc. Fundación Wikimedia. *Wikipedia, la enciclopedia libre*. 3020. URL: https://es.wikipedia.org/wiki/Flask.

[75] Plotly. *Introduction | Dash for Python Documentation | Plotly*. 2020. URL: https://dash.plotly.com/introduction (visitado 30-08-2022).

[76] Inc. Fundación Wikimedia. *Plotly - Wikipedia*. URL: https://en.wikipedia.org/wiki/Plotly (visitado 16-06-2022).

[77] Inc. Fundación Wikimedia. *Wikipedia, la enciclopedia libre*. 3020. URL: https://es.wikipedia.org/wiki/React.

[78] Jhosep Ramirez. *Arquitectura física de una aplicación web by Jhosep Ramirez*. URL: https://prezi.com/9abmgjbxhsgi/arquitectura-fisica-de-una-aplicacion-web/ (visitado 16-06-2022).

[79] Anaconda. *Anaconda | Enterprise DS Platform*. URL: https://www.anaconda.com/products/enterprise (visitado 31-08-2022).

[80] Jupyter. *Jupyter Notebook: Una introducción – Real Python*. URL: https://realpython.com/jupyter-notebook-introduction/ (visitado 31-08-2022).

[81] Monografias.com. *Intervalos de confianza con Z y t de Student empleando Excel, Winstats y GeoGebra - Monografias.com*. URL: https://www.monografias.com/trabajos97/intervalos-confianza-distribucion-normal-y-t-student-empleando-tics/intervalos-confianza-distribucion-normal-y-t-student-empleando-tics (visitado 06-09-2022).

[82] Gustavo Hideo. *10 Essential Numerical Summaries in Statistics for Data Science (Theory, Python and R) | by Gustavo Hideo | Towards Data Science*. URL: https://towardsdatascience.com/10-essential-numerical-summaries-in-statistics-for-data-science-theory-python-and-r-f3ee5e0eca32 (visitado 15-09-2022).

[83] RTVE. *El Coronavirus: Gráficos, Mapas y Datos del COVID-19 - RTVE.es*. 2021. URL: https://www.rtve.es/noticias/coronavirus-graficos-mapas-datos-covid-19-espana-mundo/ (visitado 05-09-2022).

[84] D Kelmansky. *Estadística para todos*. Vol. 7. 2. 2014, págs. 107-15. ISBN: 9789500007139. URL: https://www.estadisticaparatodos.es/taller/graficas/cajas.html.

[85] Corentin Bayette. «Images des mathématiques». En: *Noûs* (2004). URL: https://images.math.cnrs.fr/Modelamiento-de-una-epidemia-segunda-parte.html?lang=fr.