



Trabajo Fin de Grado

Mejora en la planificación del transporte no
urgente de pacientes
Better planning of non-urgent patient transportation

Autor

Arturo García Enguita

Director

Francisco José Martínez Domínguez

Escuela Universitaria Politécnica de Teruel
2020

Agradecimientos

En primer lugar, quiero agradecer a mi tutor Francisco quien me ha guiado a lo largo cada una de las etapas de este proyecto.

A toda mi familia en general, en especial a mi hermano que siempre está a mano para ayudarme y sobre todo a mis padres por animarme con mis estudios y brindarme la oportunidad de realizar este grado.

Finalmente agradecer también a mis compañeros del muy pequeño laboratorio de iNit con los que he amenizado el día a día del desarrollo del presente documento.

Mejora en la planificación del transporte no urgente de pacientes

RESUMEN

En este proyecto se optimiza una aplicación logística ya existente, NURA, siglas de Non-Urgent Routing Algorithm. Este software de enrutado facilita la coordinación de dotaciones de ambulancias para el desplazamiento de pacientes con alguna clase de tratamiento no urgente (diálisis, oncología, rehabilitación, etc.) de su hogar al centro hospitalario y viceversa.

El software recurre a una base de datos con registros de direcciones geolocalizadas y rutas, con su distancia y tiempo, que se emplean en un algoritmo genético para obtener la mejor combinación de traslados posible. Desafortunadamente, procesos ETL ajenos agravados por el incorrecto trato del bilingüismo de ciertas regiones de España han dado lugar a una colección de datos duplicados y de poca fidelidad.

Este trabajo está compuesto por dos apartados bien diferenciados, por un lado, el desarrollo de una aplicación de corrección y mantenimiento de los datos geográficos que ha de consumir el algoritmo logístico. Y por el otro un estudio estadístico de los parámetros de entrada del algoritmo para optimizar futuras ejecuciones de NURA.

La nueva aplicación, de nombre NURATools, se enfocará en la corrección automatizada de direcciones y rutas asociadas a la base de datos, entre otras herramientas básicas de gestión de usuarios. Para ello, se hace uso de los webservices de geolocalización de Cercalia y Google, que mediante un gestor de tareas en background, permiten cruzar datos con los ya existentes y verificarlos de manera automática minimizando la gestión e intervención humana.

Para el estudio de los parámetros de entrada, se realizan análisis mediante técnicas de minería de datos sobre un conjunto de más de 100.000 ejecuciones del algoritmo de enrutado, se busca seleccionar las colecciones de parámetros más idóneas para mejorar unos indicadores determinados. Estos son: tiempo de viaje de paciente, uso de ambulancias, tiempo de ejecución del algoritmo y por supuesto factibilidad de la solución propuesta.

Palabras clave: algoritmo genético, logística, ETL, validación, minería de datos.

Better planning of non-urgent patient transportation

ABSTRACT

In this project, an existing logistics application, NURA, which stands for Non-Urgent Routing Algorithm, is optimised. This routing software facilitates the coordination of ambulance crews for the movement of patients with some kind of non-urgent treatment (dialysis, oncology, rehabilitation, etc.) from their home to the hospital and vice versa.

The software uses a database with records of geolocated addresses and routes, with their distance and time, which are used in a genetic algorithm to obtain the best possible combination of transfers. Unfortunately, third-party ETL processes aggravated by the incorrect treatment of bilingualism in certain regions of Spain have resulted in a collection of duplicate and low-fidelity data.

This work is composed of two well differentiated parts, on the one hand, the development of a correction and maintenance application of the geographic data that will be used by the logistic algorithm. On the other hand, a statistical study of the input parameters of the algorithm to optimise future NURA executions.

The new application, called NURATools, will focus on the automated correction of addresses and routes associated with the database, among other basic user management tools. To do this, it makes use of the geolocation web services of Cercalia and Google, which, through a background task manager, allow data to be crossed with existing data and verified automatically, minimizing human management and intervention.

For the input parameters study, data mining techniques are used to analyse a set of more than 100,000 routing algorithm runs. The aim is to select the most suitable collections of parameters to improve certain indicators. These are: patient travel time, use of ambulances, algorithm execution time, and of course, the feasibility of the proposed solution.

Keywords: genetic algorithm, logistic, ETL, validation, data mining.

Tabla de contenido

1. Introducción.....	11
1.1. Motivación.....	11
1.2. Objetivos e hipótesis	12
1.3. Estructura de la memoria.....	12
2. Marco teórico	13
2.1. Transporte sanitario.....	13
2.2. Geolocalización y rutas.....	14
3. Estado del arte	15
3.1. Algoritmos genéticos	15
3.2. Presentación de NURA.....	17
3.3. Optimización de rutas.....	18
4. NURATools	19
4.1. Análisis	19
4.1.1. Requisitos funcionales	19
4.1.2. Requisitos no funcionales.....	20
4.1.3. Casos de uso.....	20
4.1.4. Definición de interfaces.....	21
4.1.5. Prototipos	22
4.2. Diseño UML.....	23
4.2.1. Diagramas de clases	23
4.2.2. Diagramas de actividades y secuencia	25
4.3. Implementación.....	26
4.3.1. Problemas e imprevistos.....	28
5. Optimización de NURA.....	29
5.1. Factibilidad	33
5.2. Sostenibilidad.....	36
5.3. Resultados	38
6. Conclusiones.....	39
Bibliografía	40
Anexos	41
Anexo I. Script para dar soporte a NuraTools en la BD de NURA.....	41
Anexo II. Manual de NuraTools	42

Tabla de ilustraciones y tablas

Ilustración 1. Diagrama de un algoritmo genético simple.....	15
Ilustración 2. Fenotipo y cromosomas establecidos para NURA (Fogue, y otros, 2016).	17
Ilustración 3. Diagrama de casos de uso.....	20
Ilustración 4. Ventana principal de Nuratools.	21
Ilustración 5. Ventana de edición de Nuratools.....	21
Ilustración 6. Clases básicas del modelo.....	23
Ilustración 7. Clases para proveer y validar.....	24
Ilustración 8. Clases a administrar.....	24
Ilustración 9. Persistencia.....	24
Ilustración 10. Actuación ante los tipos de la aplicación.....	25
Ilustración 11. Implementación de MVC para Java.....	26
Ilustración 12. Relación entre tiempo de viaje y servicios a realizar.	31
Ilustración 13. Impacto del número de generaciones y de individuos en el tiempo de ejecución.....	32
Ilustración 14. Gráficas de factibilidad en función del número de generaciones.	33
Ilustración 15. Relación entre tiempo total viajes y factibilidad.	34
Ilustración 16. Factibilidad en función del número de servicios.....	35
Ilustración 17. Relación entre el uso de ambulancias y los pacientes por viaje.	36
Tabla 1. Tiempos de ejecución del algoritmo de validación con n-Threads.....	27
Tabla 2. Agrupación de las zonas mediante el algoritmo esperanza-maximización.	31
Tabla 3. Factibilidad por tamaño de torneo.	32
Tabla 4. Factibilidad desglosada por código.....	34
Tabla 5. Media de pacientes por viaje y zona.....	37

1. Introducción

Actualmente ya se puede considerar a la Inteligencia Artificial como uno de los mayores campos de aplicación de las ciencias computacionales. Su capacidad para dar solución a problemas de gran complejidad la han convertido en una disciplina de estudio recurrente con cada nuevo avance a nivel hardware. Durante estos últimos 40 años se ha logrado poner en práctica técnicas, ya enunciadas a nivel teórico, cada vez más potentes y sofisticadas.

La necesidad diaria de dar respuesta a problemas logísticos de gran envergadura, para trazar las más optimas rutas de reparto o suministro, ha propiciado el desarrollo de algoritmos evolutivos, que cuentan con la capacidad de generar múltiples soluciones, adaptándose y autoevaluando sus propios cálculos para aportar un resultado en un modesto periodo de tiempo, lo que en definitiva suponen una ventaja estratégica para cualquier empresa.

1.1. Motivación

El principal objetivo de este proyecto es mejorar la eficiencia de una aplicación que ya se encuentra en producción, NURA, siglas de Non-Urgent Routing Algorithm. Es un software para el trazado de rutas de ambulancias sin urgencia, es decir, cuyos servicios suelen consistir en el traslado de pacientes no hospitalizados, que precisan de alguna clase de tratamiento clínico, pero no disponen de los medios necesarios para desplazarse a un centro médico.

Mediante un algoritmo genético, subtipo de los evolutivos, NURA recurre a una base de datos de rutas y geolocalizaciones para llevar a cabo los cálculos necesarios, generando las posibles combinaciones de traslados entre las distintas ambulancias de las que se disponga. Desafortunadamente, la información presente en dicha base de datos es en algunos casos de poca calidad. Por un lado, la existencia de registros repetidos, el bilingüismo innato de regiones de España como Cataluña o Valencia entre otras, da lugar a equívocos al ser posible denominar una misma dirección con dos lenguas. A esto hay que añadir posibles fallos humanos en la introducción de datos o una falta de supervisión en la carga automática de los mismos.

Para terminar, hay que añadir que se dispone de una muestra de unos 10.000 registros de datos pertenecientes a un periodo desde enero a mayo de 2019, con información de distintas ejecuciones de NURA para casos reales a lo largo de la costa catalana, entre las que se incluyen las ciudades de Barcelona y Gerona, junto a otros municipios circundantes. Esto supone un ideal punto de partida para llevar a cabo un estudio estadístico sobre el que, con técnicas de Big Data, optimizar el algoritmo y establecer una serie de reglas o consejos a seguir para nuevos problemas similares.

1.2. Objetivos e hipótesis

El mayor problema a día de hoy existente en NURA es su base de datos, con información de cuestionable calidad que en algunos casos es incompleta, y que impide llevar a cabo la correcta ejecución del algoritmo genético. Desgraciadamente, no se tiene constancia del uso de ningún tipo de herramienta Extract Transform Load (ETL), los nuevos registros de direcciones y servicios son añadidos manualmente sin protocolos bien definidos. Por lo tanto, la solución pasa por crear una nueva herramienta para administrar, corregir y validar los datos, previo a su uso por parte del algoritmo.

Esa nueva herramienta será NURATools, una aplicación que cruce los datos de geolocalización presentes en la BD, cuyo único proveedor es Cercalia, con los de Google o terceros. Este contraste permitirá verificar instantáneamente los datos que presenten poca desviación, dejando en manos de un usuario una futura verificación definitiva, o la elección de un dato u otro en caso de discrepancia entre las fuentes, llegando a la introducción manual del mismo como último recurso.

Solventado el tema de la calidad de los datos, otra posible vía de mejora pasa por optimizar los parámetros de entrada de NURA. Para ello, valiéndose de la muestra de casos reales, se preparará un script para realizar múltiples ejecuciones sobre cada caso alterando en todo momento uno de los parámetros del algoritmo con la finalidad de establecer las relaciones existentes entre los ya mencionados parámetros de entrada y los datos de salida. Esto permite en última instancia, establecer unas políticas de actuación para informar al usuario acerca de qué pasos a seguir en su labor de seleccionar la mejor configuración de cara a futuras ejecuciones.

1.3. Estructura de la memoria

El presente documento consta de los siguientes apartados:

- *Marco teórico*: en esta primera sección se hace una breve introducción al sistema de transporte sanitario, realizando una breve explicación de su funcionamiento. También se listarán los distintos proveedores de servicios de geolocalización.
- *Estado del arte*: esta sección contiene los conocimientos necesarios para comprender el funcionamiento básico de un algoritmo genético como NURA.
- *NURATools*: Este apartado engloba todos los pasos de la planificación del proyecto software de NURATools en sus fases de análisis, con sus requisitos y casos de uso para establecer las funcionalidades básicas a definir; diagramas de diseño en lenguaje UML; y detalles e imprevistos encontrados durante la implementación del mismo.
- *Optimización de NURA*: Este otro apartado se centra en el análisis estadístico de datos reales de ejecuciones de NURA para su optimización mediante Big Data.
- *Conclusiones*: Última sección que incluye una reflexión acerca de todo el proyecto realizado y de la relevancia de la analítica de datos.

2. Marco teórico

Previo a la realización de este proyecto, se ha realizado un breve trabajo de investigación acerca de las peculiaridades del sistema de transporte sanitario y de las distintas soluciones disponibles en el ámbito de la geolocalización.

2.1. Transporte sanitario

El transporte sanitario consiste en el traslado de pacientes hacia un centro sanitario o de vuelta a su domicilio, se realiza en un vehículo, ambulancia, con distintos grados de medicalización según la tipología o urgencia del traslado.

El más común es el transporte sanitario no urgente (TSNU). Se orienta a pacientes que por causas exclusivamente clínicas están incapacitados para desplazarse en los medios ordinarios de transporte a un centro médico, o a su domicilio tras recibir la atención sanitaria correspondiente, y que pueden precisar o no de la misma durante el trayecto.

NURA se centra únicamente en el TSNU programado, en el que los pacientes tienen que acudir a sus citas sanitarias en una fecha señalada y en el que se conoce el traslado con bastante tiempo de antelación, al menos 24 horas. El ejemplo más común, se encuentra en los servicios de diálisis, oncología, rehabilitación, etc.

Los vehículos a emplear suelen ser de tipo A1 o A2, es decir ambulancias de transporte convencional enfocadas al traslado individual o colectivo, y, que se pueden equipar con distintas configuraciones de asientos, sillas de ruedas o camillas (Verdugo & López, 2020).

La dificultad a la hora de la planificación del TSNU reside en la necesidad de cuadrar todos los horarios, tanto de traslado y recogida de los pacientes, como de la salida y vuelta a base del vehículo para la jornada de trabajo de la dotación de la ambulancia. En algunas ocasiones resulta imposible establecer una planificación, normalmente porque se cuenta con un número muy reducido de vehículos con respecto al número de servicios a realizar. Por el contrario, en la mayoría de casos, aunque se logre una solución realizable es muy complicado comprobar si dicha solución es la más idónea, esto se debe a que el reparto de las ambulancias está determinado por variaciones con repetición condicionadas por el número de servicios, lo que viene a ser $VR_{a,s} = a^s$.

Para comprender la magnitud de estos problemas, de orden exponencial, ante un caso que presente 60 servicios y se disponga de 6 ambulancias existen aproximadamente 5×10^{46} soluciones distintas, no todas útiles. El análisis manual del conjunto de soluciones queda totalmente descartado, así como, cualquier sistema basado en algoritmos tradicionales que deban analizarlo en su totalidad.

2.2. Geolocalización y rutas

La geolocalización se entiende comúnmente como la capacidad de un dispositivo de conocer su ubicación geográfica, sin embargo, en el contexto de este documento su acepción es la de transformar una dirección textual en geocoordenadas (duplas de latitud y longitud), es decir, localizar la dirección en un mapa de forma inequívoca.

En lo referente a las rutas, estas son polilíneas formadas por geocoordenadas, un vector en el que, conocidas las características de la vía que recorren, se puede extraer el tiempo necesario para recorrerlas.

Geocoordenadas y rutas son extraídas de mapas, como el que OpenStreetMap pone a disposición pública y que en estos momentos cuenta con unos 1.2 Terabytes de datos. Sin embargo, dado que para su manipulación y consulta es necesario todo un servidor dedicado, lo más común es recurrir a terceros.

Los servicios de geolocalización empleados han sido Cercalia, que ya se empleaba con NURA, y Google. Ambos sistemas ponen a disposición del usuario webservices para consultar y consumir datos en, por lo general, formatos JSON o XML.

Google cuenta con la posibilidad de añadir penalizaciones temporales en función del tráfico actual, lo que permite conocer el tiempo real para recorrer una ruta en una hora determinada. Además, a diferencia de Cercalia que establece un formato muy estricto para sus consultas, su sistema hace uso de técnicas de procesado de lenguaje natural, lo que le permite emplear directamente las direcciones textuales.

La mayor fortaleza de Cercalia reside en que es algo más preciso que su análogo a la hora de geolocalizar direcciones en municipios de tamaño reducido. Dado que el TSNU encuentra a parte de sus usuarios en dichas villas, que suelen carecer de un centro médico local o escasamente equipado, se les dará mayor importancia a los datos suministrados por Cercalia.

Existen otras soluciones de geolocalización, algunas gratuitas, aunque muy restringidas en cuanto al número de peticiones por segundo o a la existencia de límites diarios.

3. Estado del arte

3.1. Algoritmos genéticos

Como ya se ha mencionado, NURA hace uso de un algoritmo evolutivo, que tiene como objetivo la resolución de sistemas complejos, con un rango de soluciones inabarcable desde las técnicas tradicionales, pero aproximable mediante la aplicación de teorías darwinistas de selección natural. En los que, dado un entorno, metáfora del problema, y una población de individuos, abstracción de las posibles soluciones, éstos compiten por su reproducción. La aptitud de cada nuevo individuo determinada por el entorno, junto al paso de generaciones, termina imponiendo determinados genes en la población, no necesariamente los mejores, simplemente aquéllos que les permitan sobrevivir en dicho entorno (Fogue, y otros, 2016).

Al igual que el resto de ramas de la Inteligencia Artificial se puede establecer sus orígenes teóricos en torno a la mitad del siglo XX. Entre las discretas menciones por parte de Allan Turing a esta tecnología y con especial mención a los trabajos del biólogo Alexander S. Fraser sobre el uso de sistemas biológicos en un computador, se sentarían las bases de lo que acabaría siendo el algoritmo genético de parte de Holland y Goldberg entre las décadas de los 70 y los 90 (Melián, Moreno, & Moreno, 2009).

El algoritmo genético, que al final es una simple extensión de lo ya enunciado, se amplía mediante el uso de criterios de selección probabilísticos añadiendo la diversidad necesaria para no obviar soluciones potenciales. En lo relativo a la definición de cada individuo, se añaden los conceptos de fenotipo y genotipo o cromosoma. El fenotipo es el conjunto de datos que conocemos de cada individuo, necesarios para calcular su aptitud; mientras que, el cromosoma es un subconjunto del anterior, y consiste en una cadena de genes concretos, tradicionalmente un vector binario (Wikipedia, 2020).

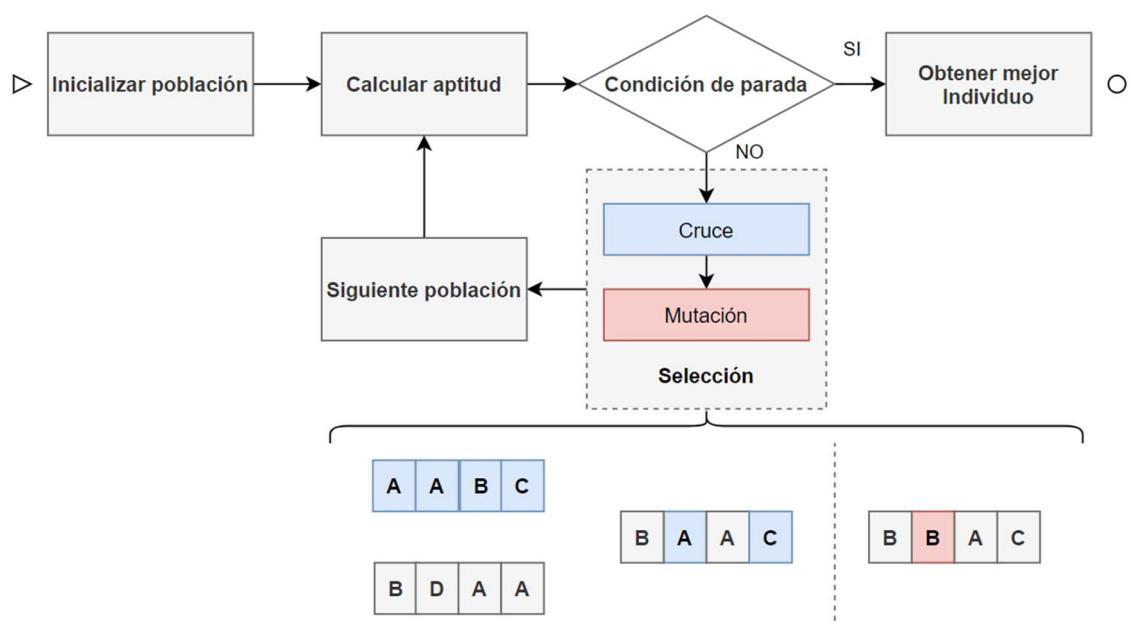


Ilustración 1. Diagrama de un algoritmo genético simple.

En lo relativo al funcionamiento y como se observa en la *Ilustración 1*, el primer paso es generar una población inicial, de forma aleatoria o sirviéndose de alguna hipótesis. Hasta que se cumpla la condición de parada, se calculará la aptitud de cada individuo para seleccionar a los reproductores:

- *Por Torneo*: Se elige al azar un conjunto y de ellos se toma el mejor.
- *Orden lineal*: Se ordenan de mejor a peor y se escogen los mejores.
- *Aleatoria pura*: Se escogen pares totalmente al azar.
- *Procedimiento de la ruleta*: Se asignan probabilidades y se eligen padres de acuerdo a ellas.
- *Diverso/aleatorio*: Se escogen unos padres al azar y cada uno de ellos se empareja con el individuo más diferente a él.
- *Diverso/elitista*: Se escogen los mejores y cada uno de ellos se empareja con el individuo más diferente a él.

A la hora de realizar un cruce también se dispone de distintos operadores, se puede recurrir a los unipuntuales o n -puntuales en los que se selecciona una o más posiciones del cromosoma para recombinar sus subdivisiones, o simplemente dejarlo todo en manos del azar intercambiando n posiciones (Melián, Moreno, & Moreno, 2009).

Para terminar con este ciclo, la mutación es de vital importancia, pues evita que se pueda terminar rápidamente con un único genoma dominante, abriendo siempre la puerta a una posible nueva mejora de la reserva genética. Por lo general, suele ser un mecanismo basado en un fuerte componente aleatorio.

Para conformar la nueva generación, se llevará a cabo un reemplazo parcial o total de la anterior con mecanismos de selección similares a los anteriormente comentados. Una vez se cumpla la condición de parada se extrae al individuo de mayor aptitud, su fenotipo contiene una de las posibles (o no) soluciones al problema (Fogue, y otros, 2016).

3.2. Presentación de NURA

A día de hoy, NURA se emplea para dar soporte logístico a más de cien bases de ambulancias, lo que se traduce en unos dos mil vehículos capaces de dar servicio a cerca de 4 millones de pacientes anualmente. Permite establecer itinerarios en minutos, que serían horas manualmente, reduciendo la carga de trabajo depositada en los planificadores. Esto les permite aprovechar mejor su tiempo para simplemente escoger la opción más adecuada de las sugeridas por la aplicación.

Además, a lo largo de su periodo de producción se ha apreciado una mejora importante en la optimización de los tiempos de traslado de los pacientes, mejorando la calidad del servicio, y de paso, reduciendo el impacto medioambiental de las ambulancias.

A nivel técnico, se puede definir a NURA como un algoritmo genético no binario con selección por torneo, cruce n -puntual y reemplazo generacional. Su condición de parada viene condicionada por un número de generaciones predeterminado.

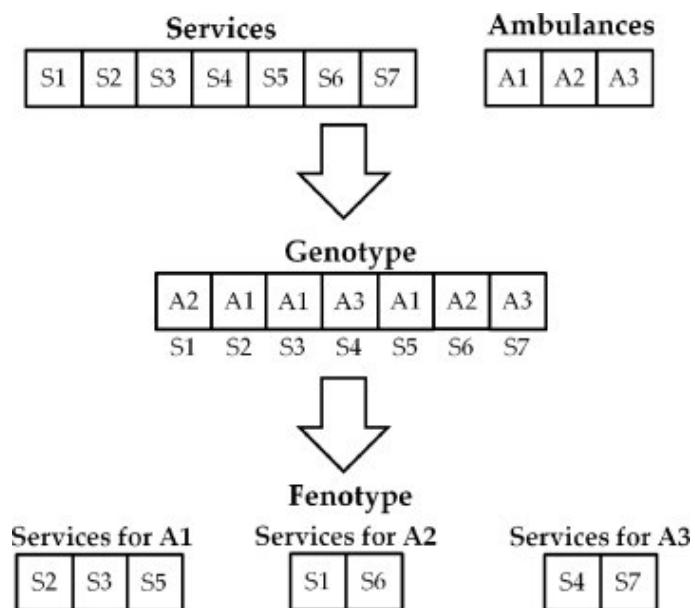


Ilustración 2. Fenotíp y cromosomas establecidos para NURA (Fogue, y otros, 2016).

Como se observa en la *Ilustración 2*, el genotipo de cada individuo queda determinado en longitud por los servicios a organizar, siendo las ambulancias asignadas a cada servicio los genes a emplear. Al generar el fenotipo se asigna a cada ambulancia los servicios a realizar con los que se calcularán su adaptabilidad. Sin embargo, dados los tiempos de desplazamiento desde la base a un servicio o entre los mismos, junto a la necesidad de depositar a los pacientes en tiempo y forma, algunos individuos tienen una adaptabilidad directamente nula, es decir son soluciones imposibles. Este el concepto de no factibilidad (Fogue, y otros, 2016).

Este concepto es de gran importancia, ya que con demasiados servicios frente a pocas ambulancias puede llegar a no existir ningún individuo factible entre todo el espacio de soluciones. Para otros casos en el que la relación servicios/ambulancia sea ajustada, se necesita de un mayor número de generaciones para una única solución, lo que implica mayor tiempo de ejecución.

3.3. Optimización de rutas

Para la optimización de NURA se va a recurrir a las herramientas empleadas para el análisis de grandes volúmenes de datos, Big Data. Aunque la muestra de la que se dispone no cuenta con el volumen necesario para ser considerada como tal, cumple con dos de los factores fundamentales de la disciplina, la variedad y la veracidad. Son datos reales suministrados por la empresa que explota NURA, en los municipios de las provincias de Barcelona y Gerona, cuyos dos núcleos urbanos difieren en términos de población, tanto en número de habitantes como en media de edad.

Asociado comúnmente al Big Data se encuentra el data mining, la minería de datos es el conjunto de técnicas heredadas tanto de la estadística como de la inteligencia artificial, más concretamente del aprendizaje automático, cuyo fin es extraer conocimiento de un conjunto de datos. Dichas técnicas o algoritmos aparecen clasificados en WEKA como:

- *Clasificadores*: como su nombre indica clasifican los datos en función de un atributo, engloba a un gran conjunto de técnicas de las que se han empleado:
 - *Regresión*: es la más sencilla, se utiliza para establecer relaciones entre dos variables con dependencia unidireccional. Consiste en el ajuste del modelo a una función (polinómica, logarítmica, exponencial...) con la que predecir comportamientos futuros.
 - *Árboles de decisión*: son un tipo de modelos de predicción basados en reglas, que categorizan una serie de condiciones para la resolución de un problema.
- *Clustering*: es un procedimiento de agrupación según criterios de distancia o similitud. Se emplea para clasificar los datos en función de una variable, creando grupos que poseen propiedades comunes.
- *Reglas de asociación*: se utilizan para establecer relaciones entre dos hechos de un conjunto de datos, en los que una determinada acción suele desencadenar específicamente otra.
- *Selección de atributos*: son el conjunto de métodos para identificar los atributos que tienen más peso a la hora de determinar si los datos son de una clase u otra.

4. NURATools

Toda esta sección incluye la información relativa a la primera gran aportación de este TFG, el proyecto y desarrollo de un software para la gestión de direcciones. En concreto se presentan las fases de análisis, diseño e implementación de la aplicación semiautomática de mantenimiento, corrección y validación de los datos geográficos que emplea NURA.

4.1. Análisis

Se busca diseñar una herramienta de fácil empleo cuyo uso sea puntual y muy reducido, operando de forma autónoma siempre que sea posible.

4.1.1. Requisitos funcionales

RQ 1. La aplicación empleará un mínimo de dos proveedores online de servicios de geocodificación, de cara a un posterior cruce y validación de los datos de la BD de NURA.

RQ 2. Dada una dirección, ésta ha de poder geolocalizarse, obteniendo posibles longitudes y latitudes.

RQ 3. Dadas dos coordenadas, que representan un servicio, se han de obtener las posibles rutas con su distancia en metros y minutos.

RQ 4. La aplicación ha de ejecutar mediante un scheduler de forma concurrente y autónoma las tareas enumeradas en los dos puntos anteriores, RQ 2 y RQ 3.

- RQ 4.1. Se dará prioridad a las tareas de las que se disponga menor información
- RQ 4.2. Se podrá parar y reanudar el proceso automático en todo momento.
- RQ 4.3. El usuario podrá solicitar la realización de una tarea con prioridad.

RQ 5. Para las múltiples coordenadas o rutas generadas se comprobará y validará la más acertada para la dirección o servicio correspondiente. Se producirá una autovalidación en función de la diferencia existente entre las posibles soluciones.

- RQ 5.1. El usuario podrá alterar los parámetros que determinan la autovalidación.
- RQ 5.2. Manualmente se podrá validar y/o editar una coordenada o servicio.
- RQ 5.3. Para un elemento validado se indicará al autor o proveedor del mismo.

RQ 6. Se ha de contar con las herramientas básicas de creación, edición y eliminación, para los usuarios, zonas y bases de NURA.

- RQ 6.1. Para cada usuario se podrán asignar unas zonas a administrar.
- RQ 6.2. Cada base estará ligada a una única zona.

4.1.2. Requisitos no funcionales

RQ 7. Se permite la modificación de la base de datos, siempre y cuando se añadan nuevas tablas o columnas. No se deben eliminar o editar las existentes.

RQ 8. Se ha de tratar de reducir al máximo en la base de datos el peso de los nuevos datos necesarios para esta aplicación.

RQ 9. El empleo de esta herramienta ha de resultar en una experiencia de usuario positiva.

RQ 9.1. Se debe de minimizar la carga de trabajo añadida.

RQ 9.2. La aplicación ha de poder operar sin supervisión alguna (RQ 4).

RQ 10. El sistema proporcionará mensajes de error informativos orientados para un usuario final.

RQ 11. El sistema contará con un manual de usuario correctamente estructurado.

4.1.3. Casos de uso

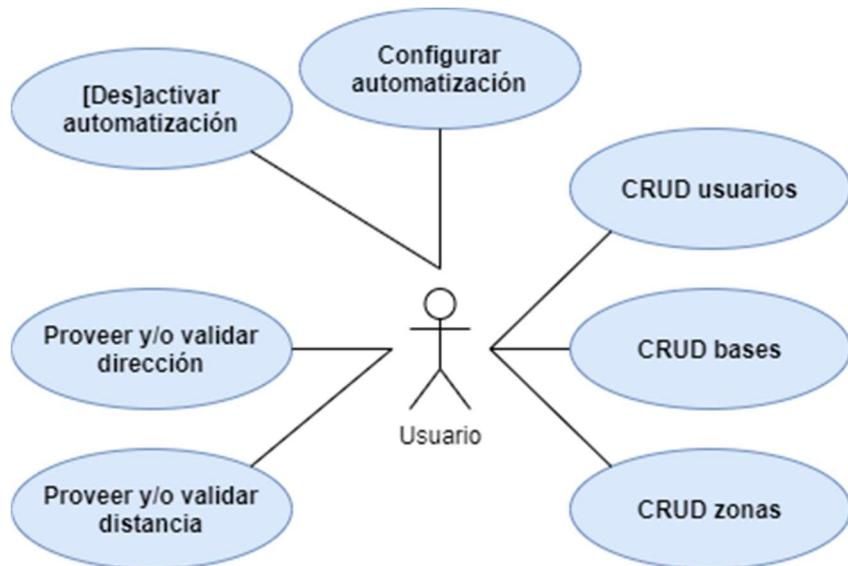


Ilustración 3. Diagrama de casos de uso.

La *Ilustración 3*. Diagrama de casos de uso define un único usuario para la aplicación, con la capacidad para administrar a otros usuarios, bases y zonas de la base de datos de la aplicación principal mediante las herramientas básicas de edición. No obstante, no puede crear nuevas direcciones ni servicios, pero sí que puede actualizar o alterar los datos de las presentes y futuras. Finalmente cuenta con la capacidad de configurar el mecanismo y los criterios de automatización.

4.1.4. Definición de interfaces

Dada la necesidad de alterar múltiples registros, la mayor parte de las ventanas de la interfaz consistirá en una tabla ordenada como la de la *Ilustración 4*, en la que aparecerán ordenados por relevancia, y colores, aquellos datos que necesiten mayor atención. Se señala en rojo los datos que no puedan ser utilizados por NURA y requieran de atención inmediata, naranja para aquellos que presenten alternativas que el sistema automático no haya podido resolver y verde para los que no haya ninguna discrepancia entre los datos de los proveedores. Se dejan sin color aquéllos que no hayan sido todavía manipulados por el scheduler.

NURATools								
Direcciones	Servicios	Usuarios	Bases	Configur				
Tabla de Direcciones								
Estado	Via	Nombre via	Nºm.	Municipio	Provincia	C.P.		
Sin resultados	C	dels girasols	144	Vilta				
Sin Validar	C	de los girasoles	133	Villalta	Barcelona	08323		
Validado	C	de los girasoles	102	Villalta	Barcelona	08323		
Sin alternativas	C	de los girasoles	93	Villalta	Barcelona	08323		
Sin alternativas		
Sin alternativas		
Sin alternativas		

Ilustración 4. Ventana principal de Nuratools.

Ventana de edición

1 Via de los girasoles

Municipio

Provincia 08323

2 Alternativas

1) 41.2334, 2.7324 - [Cercalia](#)
 2) 41.2345, 2.7325 - [Google](#)

3

4 Manual

Ilustración 5. Ventana de edición de Nuratools.

Al pulsar en cualquier opción de la tabla de la *Ilustración 4* se despliega la ventana de edición definida en la *Ilustración 5*. La ventana de edición será la misma para direcciones y servicios, con las modificaciones oportunas, contando con una sección para corregir cualquier error en los datos textuales (1), seguido de un listado para las alternativas (2). No obstante, se pondrá a disposición del usuario un mapa interactivo (3) para que con un simple clic pueda validar el elemento. En el caso de que se quisiese introducir manualmente la coordenada, se dará dicha alternativa mediante un cuadro de texto (4).

4.1.5. Prototipos

Con el fin de facilitar el desarrollo de la aplicación, se ha subdividido la implementación de la misma en 4 prototipos con fechas estimadas de finalización. Los prototipos son los siguientes:

- *Prototipo 1 (30/03/2020 – 2 meses):*
Implementación de las clases básicas para el funcionamiento del mecanismo de consumo de webservices. Además, desarrollo del sistema concurrente de autovalidado.
- *Prototipo 2 (27/04/2020 – 1 mes):*
Creación de la interfaz para dar soporte a lo desarrollado, tablas de direcciones y servicios, junto al sistema de mapas de OpenStreetMaps. Se añadirá el código oportuno para dar al usuario la posibilidad de activar/desactivar el sistema background de autovadillado.
- *Prototipo 3 (25/05/20 – 1 mes):*
Ampliación de la interfaz para las tareas de administración de los usuarios y bases de ambulancias definidas para NURA. Se incorporarán sistemas de Quality of Life (QoL) para poder deshacer y rehacer acciones.
- *Prototipo 4 (29/06/2020 – 1 mes):*
Prototipo de respaldo para detalles y optimizaciones finales. También, creación de ficheros para la persistencia de las opciones de configuración.

4.2. Diseño UML

4.2.1. Diagramas de clases

El apartado de corrección de direcciones y rutas, cuenta con las dos clases principales enunciadas en la *Ilustración 6*, por un lado, las *validables*, pueden ser una coordenada o una ruta, pudiendo ser editadas y validadas. Por el otro, las clases *providables* o suministrables, que son las que realmente proveen los webservices de geolocalización: una dirección completa con una coordenada y sus alternativas o un servicio, que son dos coordenadas con su correspondiente ruta y demás alternativas.

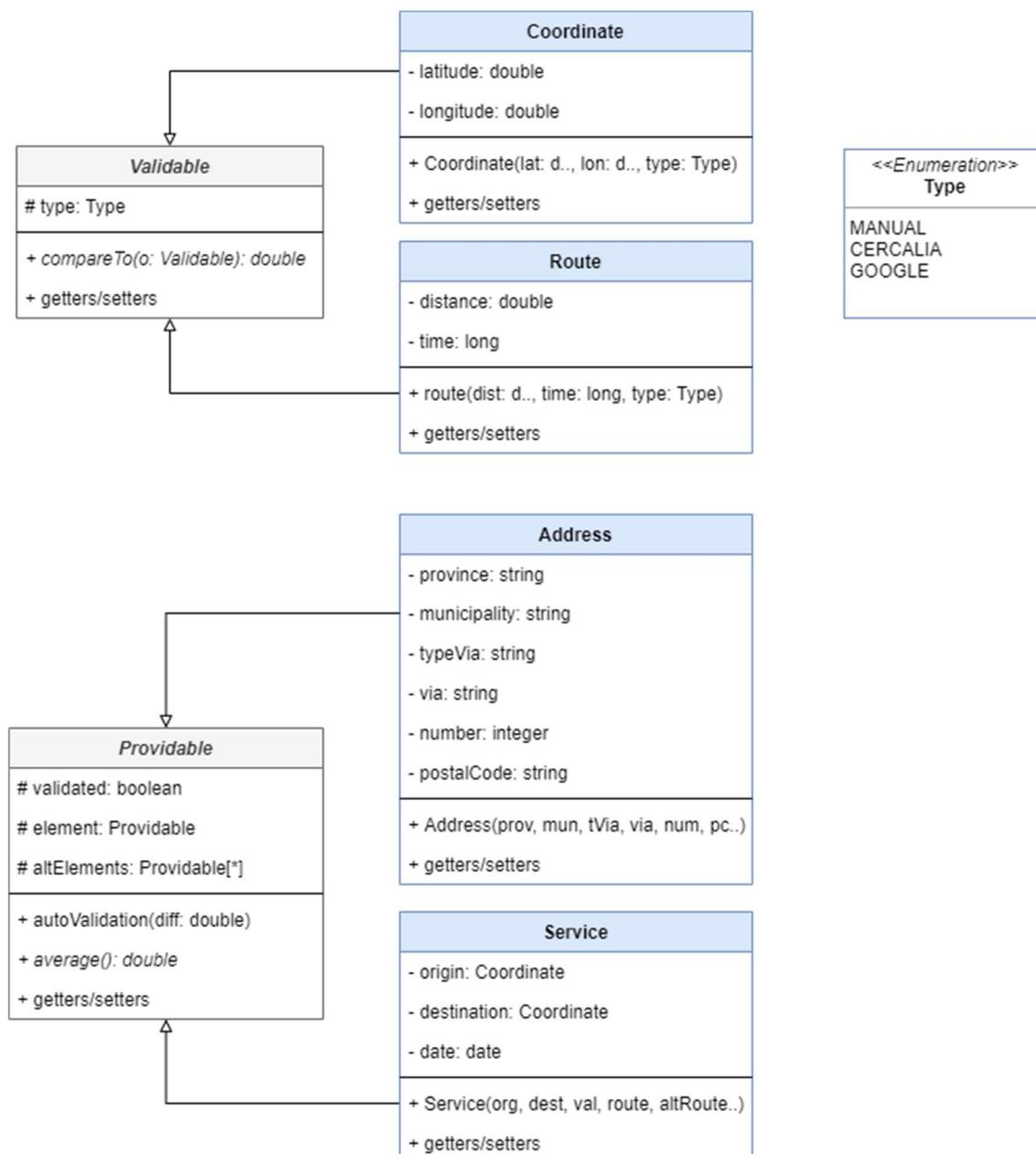


Ilustración 6. Clases básicas del modelo.

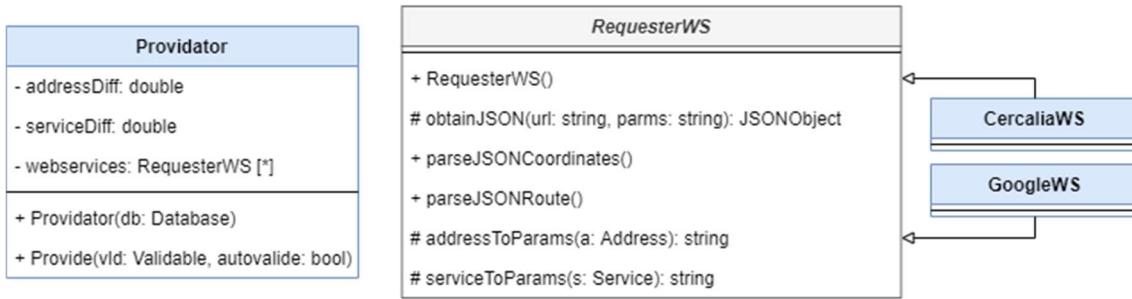


Ilustración 7. Clases para proveer y validar.

En la *Ilustración 7*, se cuenta con la clase *Providator*, es la encargada de gestionar los parámetros de configuración de la automatización, alojados previsiblemente en un fichero de texto, y también de realizar las llamadas oportunas a los webservices de los proveedores, ya sea desde un thread en background o por una solicitud del usuario. La clase *RequesterWS* ha de ser extendida para cada nuevo proveedor que quiera emplearse, añadiendo más clases necesarias en caso de que éste solo suministrase en formato XML, se recomienda el uso de JSON al ser más ligero y por ende más rápido su parseado.

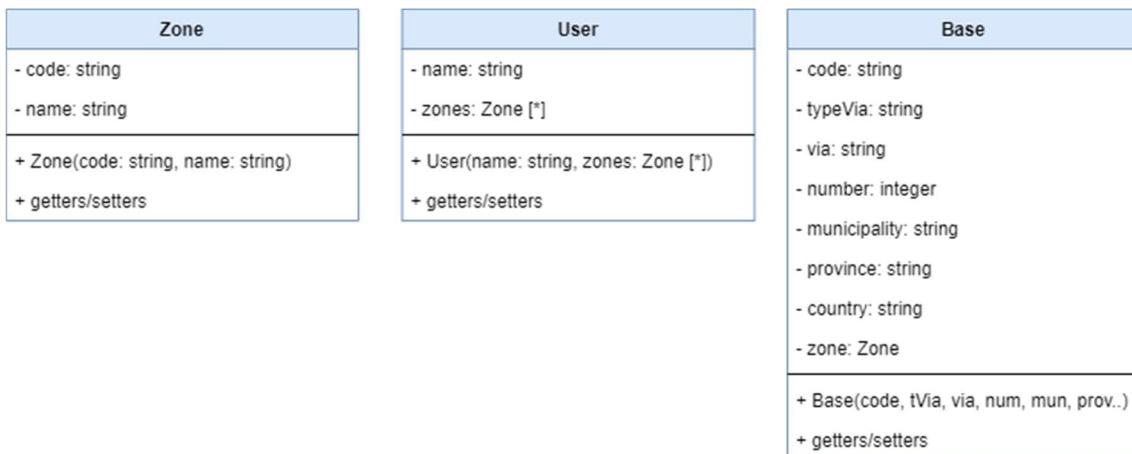


Ilustración 8. Clases a administrar.

En la *Ilustración 8* se muestran las otras clases necesarias, que simplemente han de disponer de un interfaz para realizar las oportunas operaciones de edición. Para finalizar con este apartado, la *Ilustración 9* indica qué clases han de ser almacenadas para usos futuros.

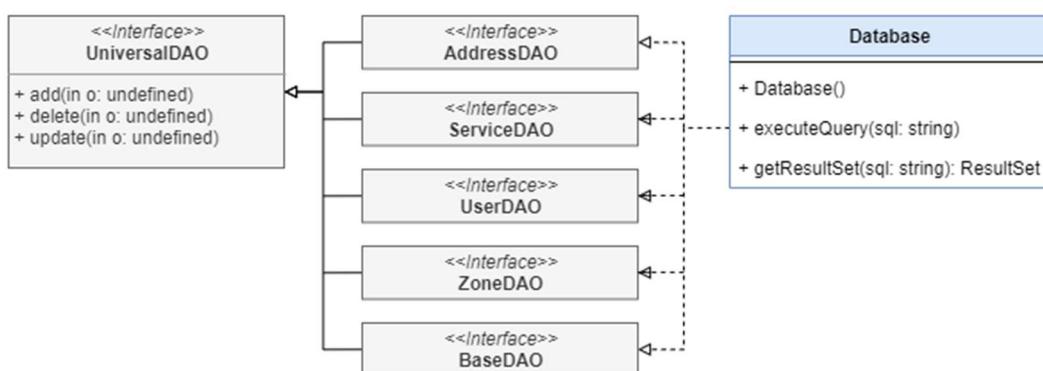


Ilustración 9. Persistencia.

4.2.2. Diagramas de actividades y secuencia

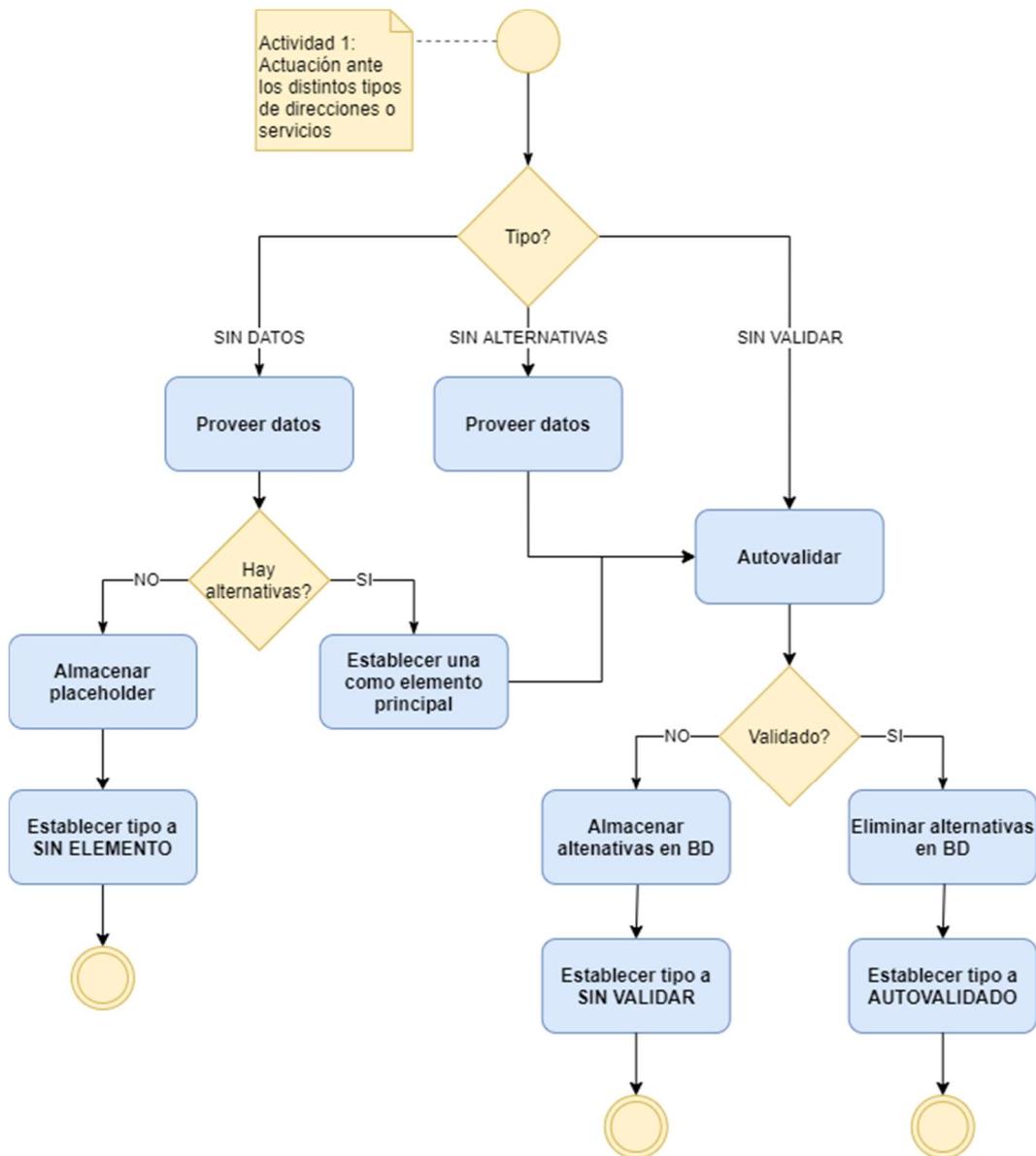


Ilustración 10. Actuación ante los tipos de la aplicación.

En la *Ilustración 10*, se define la actuación del sistema en segundo plano para la provisión y validación de coordenadas y rutas a direcciones y servicios. Se pueden dar tres escenarios, que no se disponga de ningún dato y que los proveedores no hayan podido generar ninguno adicional, es un caso bastante raro, pero en el que es necesaria la intervención urgente por parte del usuario. Y para las otras dos situaciones en las que se cuente con un elemento bien definido se puede llevar a cabo o no una autovalidación eliminando las alternativas de la base de datos, aunque se tratan de mantener en memoria principal por si el usuario quisiese comprobar manualmente la decisión del algoritmo de autovalidación, antes de cerrar la aplicación.

4.3. Implementación

Se comienza por añadir 3 nuevas tablas a la base de datos compartida:

- *geolocalizadores*: Listado de los proveedores de coordenadas y rutas.
- *coordenadasalt*: Tabla de coordenadas alternativas de una dirección.
- *duracionviajesalt*: Tabla de rutas alternativas de un servicio.

Y alterar las tablas de direcciones y servicios para indicar su estado y tipo de proveedor, añadiendo además un trigger para cada tabla de cara a facilitar el código en la aplicación. En el *Anexo I* se adjunta un script con todas las modificaciones realizadas.

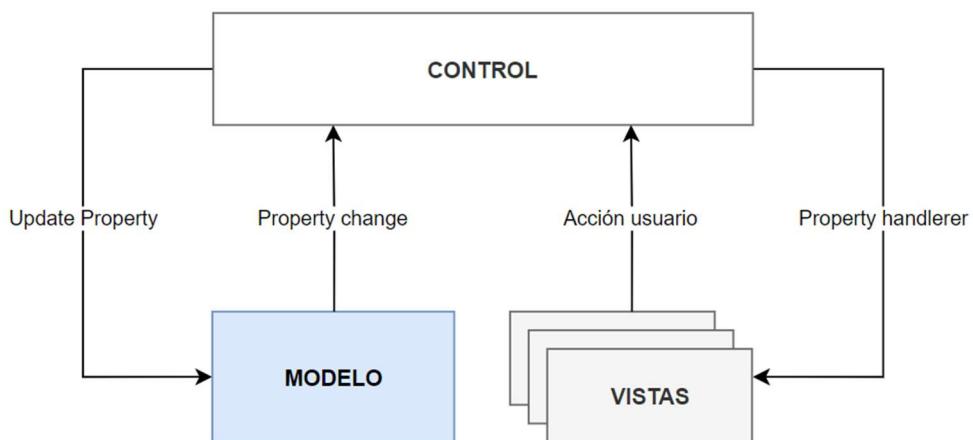


Ilustración 11. Implementación de MVC para Java.

Para la arquitectura de la aplicación se ha optado por un patrón MVC. En la *Ilustración 11* se puede ver la implementación de Java, basada en eventos del tipo *PropertyChange*, en los que el usuario al interactuar con la interfaz llama a un método del modelo a través del controlador que al finalizar dispara un evento que el controlador reenvía a cada uno de los listeners de las vistas registradas para actualizar la información en el interfaz gráfico.

Esta arquitectura, en concreto, permite trabajar con múltiples ventanas con alguna clase de información compartida. Se ha considerado necesaria para este proyecto dado que al editar elementos en ventanas secundarias se pueden producir cambios en los datos a modificar, siempre que el proceso background de autovalidado esté por supuesto en funcionamiento. Esta implementación facilita la comunicación y evita la necesidad de referencias hacia una ventana padre, simplemente al crear una ventana ésta se registra en el controlador.

En lo referente al mecanismo de autovalidado en segundo plano se optó en un principio por implementarlo mediante un único thread debido a las restricciones que establecen algunos proveedores en lo relativo a número de peticiones por segundo. Para la versión final de la aplicación se cambió a un ejecutor de servicios encolado. Éste busca ejecutar el mayor número de hilos según las restricciones establecidas por cada webservice.

Tabla 1. Tiempos de ejecución del algoritmo de validación con *n*-Threads

1-Thread		4-Threads (Executor Service)		
	Tiempo (s)	Inicio (ms)	Fin (ms)	Tiempo (s)
	1,13	1593249084504	1593249085621	1,12
	1,03	1593249091144	1593249092378	1,23
	1,23	1593249092348	1593249093396	1,05
	1,18	1593249093779	1593249094914	1,14
	1,26	1593249095722	1593249096675	0,95
	1,22	1593249095870	1593249096900	1,03
	1,27	1593249095570	1593249096767	1,20
	1,22	1593249095626	1593249096784	1,16
	1,20	1593249096941	1593249098264	1,32
	1,17	1593249095670	1593249096893	1,22
	1,13	1593249095349	1593249096423	1,07
	1,24	1593249095770	1593249096744	0,97
	1,32	1593249095803	1593249096926	1,12
	1,15	1593249095259	1593249096557	1,30
	1,10	1593249095515	1593249096657	1,14
	0,89	1593249095749	1593249096936	1,19
	1,17			1,14

Como se observa en la *Tabla 1*, la media de tiempo requerido por operación de autovalidado es bastante similar, sin embargo, en el caso de los 4-Threads al ser operaciones concurrentes, la media real es la diferencia entre la fecha de finalización del último servicio y la de comienzo del primero para el número de operaciones realizadas:

$$\frac{1593249096,936 \text{ s} - 1593249084,504 \text{ s}}{16 \text{ operaciones}} = 0,86 \text{ s}$$

De esta forma, se logra una mejora del 26% al pasar de una media de 1,17 s. por petición a otra estimada de 0,86 s. para un mínimo de cuatro threads. La media real es estimada, dado que los threads de Java son virtuales y en ocasiones se añaden algunos pequeños retardos que escapan totalmente al control del programador.

4.3.1. Problemas e imprevistos

Uno de los problemas encontrados es derivado del uso de cadenas de caracteres como clave primaria en la tabla que almacena los datos de una dirección. Java, el lenguaje empleado para la implementación de esta herramienta, tiene otras políticas de Collation con respecto a MySQL, la base de datos compartida, en lo referente a la comparación de strings, que se pueden resumir en:

- *Case sensitive*: Se diferencia entre mayúsculas y minúsculas, p. ej.: A ≠ a.
- *Accent sensitive*: Se diferencia entre un vocablo y sus derivaciones con acentos, virgulillas o cedillas entre otros, p. ej.: a ≠ á, n ≠ ñ, c ≠ ç.
- *Kana sensitive*: Se diferencia entre los caracteres Hiragana y Katakana de los alfabetos nipones.
- *Width sensitive*: Se diferencia entre caracteres de 2-byte y 1-byte, p. ej.: ² ≠ 2

Estas políticas se pueden solicitar a la base de datos mediante una consulta a sus propiedades, y parseando la respuesta establecer un Collator en el lenguaje de alto nivel que se emplee. Sin embargo, por razones de diseño, fundamentalmente para reducir dependencias, se ha optado por definirlo en un fichero de configuración.

Algo similar también ocurre para el caso de la tabla que contiene los servicios, se emplea como clave primaria tipos decimales. Esto obliga a trabajar con decimales precisos con redondeo hacia arriba (por defecto en MySQL), ya que el mínimo error en un decimal puede crear nuevos registros indeseados.

En lo relativo a los proveedores, para la comunicación con los webservices se emplea HTML, lo que supone UTF-8 y la necesidad de castear los caracteres especiales, tales como espacios, para garantizar la comunicación. Este fallo ha sido considerablemente difícil de detectar puesto que todos los navegadores modernos admiten UTF-16 en su barra de búsqueda y realizan la conversión de forma totalmente transparente al usuario al realizar la navegación.

Sin embargo, el mayor problema viene en cuanto a la recepción de datos. Uno de los proveedores ha decidido emplear JSON con claves duplicadas. El estándar EMCA-404 que especifica la sintaxis de un documento de este tipo, no hace ninguna mención en lo referente a si las claves duplicadas son válidas o no. Esto ha dado lugar a que la mayoría de implementaciones de librerías JSON no acepten claves duplicadas y que no exista un estándar de facto entre las pocas que las admiten. Por lo tanto, se recomienda evitar su uso siempre que sea posible, o emplear en su lugar un vector. Como la librería que se ha elegido para la aplicación pertenece al primer grupo, se ha tenido que capturar la excepción de formato correspondiente y optar por ignorar los valores de toda clave repetida.

5. Optimización de NURA

De esta sección en adelante, se trata la segunda gran aportación de este TFG, es decir, todo lo relativo al análisis del algoritmo genético mediante WEKA, herramienta Java freeware de data mining. Para obtener un dataset lo suficientemente representativo se han realizado unas 100.000 ejecuciones de NURA, alterando los siguientes parámetros del algoritmo:

- *numIndividuos*: Población del algoritmo genético cuyo genoma representa una posible solución (factible o no).
- *numGeneraciones*: Iteraciones del algoritmo.
- *tamanoTorneo*: Número de individuos a competir para la procreación de la próxima generación.
- *tasaRecombinacion*: Porcentaje de genes a intercambiar entre los progenitores.
- *tasaMutacion*: Porcentaje a aleatorizar del genoma de un nuevo individuo.
- *generationalGap*: Proporción de cromosomas a reemplazar por la próxima generación.
- *inicioUsoMemetismo*: Comienzo del memetismo, transferencia de genes, de cara a evitar la convergencia que pueda omitir una solución más óptima.

Con determinados casos reales (parámetros inalterables), de los que se conoce:

- *numeroServicios*: Número de traslados a realizar.
- *codigo*: Código de la zona en la que se opera.
- *fecha*: Fecha, especificando únicamente mes y día.

De cara a obtener los siguientes resultados:

- *factible*: Si al menos existe un genoma que cumple con los criterios mínimos de traslado de los distintos pacientes, es decir llega a su cita en la hora prevista y no espera demasiado a la ambulancia ni permanece en exceso en la misma.
- *penalizacionTiempo*: Media del tiempo de espera por servicio de cada paciente.
- *usoAmbulancias*: Porcentaje relativo al tiempo en que las ambulancias han estado circulando con respecto al horario de su jornada.
- *tiempoTotalViaje*: Valor total del tiempo invertido por cada ambulancia en su desplazamiento.
- *pacientesViaje*: Media de pacientes transportados por viaje de ambulancia.
- *tiempoEjecucion*: Tiempo en minutos necesarios para la ejecución del algoritmo
- *generacionesAsintota*: Generación con la mejor solución, con mayor fitness entre todo el conjunto, pero no necesariamente factible.
- *primeraGeneracionFactible*: Primera generación en la que se cuenta con un individuo factible.

El principal objetivo del estudio es mejorar la calidad del servicio, minimizando el tiempo de transporte de los usuarios. También es relevante maximizar el tiempo de uso de las ambulancias para limitar su número, con el impacto en el beneficio ambiental y económico que ello supone, siempre y cuando se mantengan los estándares de calidad del servicio. Y, por último, se pretende incrementar la factibilidad y reducir el tiempo necesario para calcular soluciones mejorando la QoL para los usuarios de la aplicación.

El primer paso antes de comenzar es aumentar el tamaño máximo del heap de la máquina Java, limitado a 2048 MB, ya que dado el tamaño de la muestra podría producirse un desbordamiento del mismo al realizar alguna operación, lo que causa un cierre forzoso de la máquina Java, y de WEKA, con la correspondiente pérdida de cualquier dato generado.

Una vez que Java está listo se comienza a transformar los datos al formato *.arff* que emplea WEKA, en el que solo caben datos de tipo numérico o listado de objetos etiquetados. Se han empleado las siguientes expresiones regulares:

Regex para separar zonas de fechas:

```
,([^\,]+)([0-9]{4})  
,\1,\2
```

Regex para eliminar filas duplicadas:

```
^(.*?)$\s+?^(?=.*^$)
```

De cara a analizar los datos de entrada de los casos reales, tras un primer vistazo a los mismos se deduce que uno de los atributos de mayor importancia es el del código de zona, existen 4 zonas de actuación para las flotas de ambulancias entre los que se encuentran las dos urbes de Barcelona y Gerona junto a otras dos más que agrupan al conjunto de municipios al norte y este de la ciudad condal, a las que para clarificar se denominarán como interior y costa.

Por el contrario, la fecha asociada a los servicios, no resulta muy relevante al indicar únicamente el día y no parece existir ninguna dependencia entre la misma y los resultados, más allá de ser conscientes de que el TSNU no opera en festivos.

En la página a continuación, se cuenta con la gráfica de la *Ilustración 12* con el código de zona representado por colores. Se observan dos claras tendencias, con pendiente mayor que 1 se encuentran los casos de zonas urbanas, con una menor área de actuación y mayor densidad de centros hospitalarios, lo que resulta en traslados más cortos. Resulta curioso que, en Gerona con una séptima parte de la población de Barcelona, sus ciudadanos recurran de media cuatro veces más a este servicio. Algo quizás, atribuible a una población más envejecida o a un peor transporte público.

Las zonas rurales, por el contrario, con pendiente inferior a 1 están definidas para una mayor superficie. También se da un menor uso del servicio entre las poblaciones de costa con respecto a las de interior, aunque su diferenciación resulta más difícil que para el caso de las dos capitales de provincia.

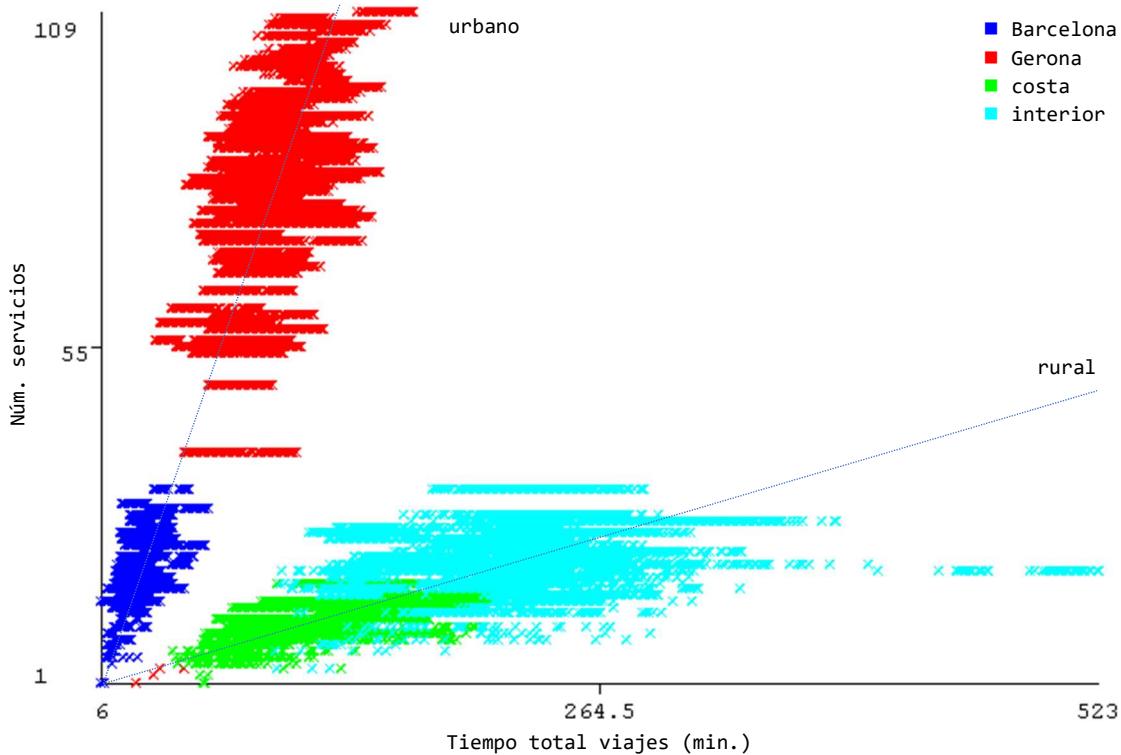


Ilustración 12. Relación entre tiempo de viaje y servicios a realizar.

Un clusterer EM (Expectation-Maximization) cuyos resultados se encuentran en la *Tabla 2*. Agrupación de las zonas mediante el algoritmo esperanza-maximización., establece la existencia de 3 grupos bien definidos. Separa el hipotético urbano en dos, otorgando un grupo concreto para cada gran ciudad, para finalmente especificar un tercero que engloba a la totalidad de las poblaciones rurales, una decisión cuestionable pues los tiempos de traslado en interior son bastante considerables con respecto a los de la costa.

Tabla 2. Agrupación de las zonas mediante el algoritmo esperanza-maximización.

Código	Cluster0	Cluster1	Cluster2
Barcelona	<u>25006</u>	1	751
Gerona	729	<u>25239</u>	33
costa	0	0	<u>26244</u>
interior	0	0	<u>30375</u>

Para futuros análisis de los datos se ha de tener en cuenta en todo momento la zona de actuación, ya que del código depende el número de vías y potenciales rutas a escoger por cada vehículo, siendo por lo general más pesada la carga computacional en zonas urbanas.

Como era de esperar, un Gain Ratio Attribute Evaluation (GRAE) confirma que el número de servicios tiene un impacto directo en el tiempo de ejecución del algoritmo. Además, permite una mayor combinación de alternativas entre las ambulancias, para las que se necesita un mayor número de generaciones si se quieren obtener resultados útiles.

Para finalizar con este apartado, queda revisar los parámetros del algoritmo. De éstos se va a comenzar por eliminar la tasa de recombinación, la tasa de mutación y el inicio de uso de memetismo. Son parámetros que no se han modificado en exceso para la muestra, sus posibles valores están muy acotados ya que de lo contrario añaden excesiva, y no deseada, aleatoriedad al algoritmo.

Como resultado se ha obtenido que de esos 3 parámetros no hay ningún dato de salida que dependa de ellos, habiendo en algunos casos según un GRAE una dependencia tan reducida que a efectos prácticos es nula.

Para el resto de parámetros, los dos fundamentales son, por un lado, el número de generaciones, ya que según el diseño de NURA es lo que establece la condición de parada, y por el otro el número de individuos, que representa las iteraciones por generación. En la *Ilustración 13* se comprueba que el producto de ambos parámetros establece el orden del algoritmo y por ende su tiempo de ejecución. No obstante, la necesidad de más o menos iteraciones, con la finalidad de obtener una solución factible, sigue estando en parte determinada por el número de servicios a realizar.

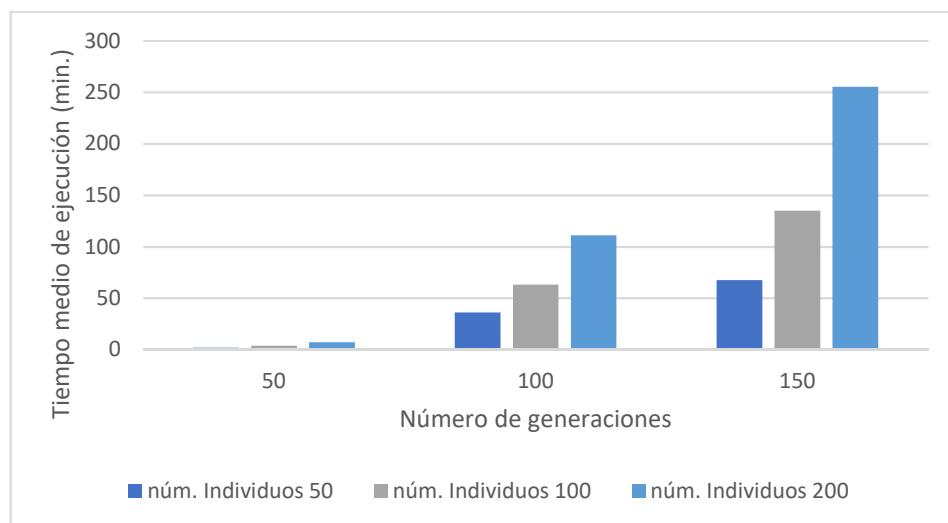


Ilustración 13. Impacto del número de generaciones y de individuos en el tiempo de ejecución.

Los dos parámetros que restan, reemplazo y tamaño de torneo tiene algo más de importancia que los eliminados, pero tampoco parecen afectar demasiado al resultado final. En términos de factibilidad, parece que se logran mayor número de positivos para un tamaño de torneo y reemplazo reducidos. Como se observa en la *Tabla 3* con mayor número de torneo, la solución converge de media más rápidamente, tanto en tiempo como de ejecución como lo que tarda en encontrar una primera generación factible. El comportamiento para el reemplazo generacional es similar.

Tabla 3. Factibilidad por tamaño de torneo.

Tamaño de torneo	Porcentaje factible	Tiempo ejecución	1º gen factible
3	0,6903	24,09 min	10
5	0,6893	23,81 min	9
10	0,6776	21,94 min	8

5.1. Factibilidad

Como ya se ha comentado, la factibilidad se da si se cumplen unos mínimos criterios de calidad del servicio, que se suele traducir en un tiempo excesivo de traslado para la distancia a cubrir, p. ej.: si la ambulancia toma un desvío para transportar a un segundo paciente. También se puede dar el caso de que no se logre llevar a tiempo a un paciente o si se le hace esperar en exceso una vez ya tratado para trasladarlo de vuelta a su hogar. Por último, las dotaciones de ambulancias tienen unos horarios, obligarles a trabajar fuera del mismo también hace la solución irrealizable.

Un GRAE establece como parámetro más relevante el número de generaciones seguido por supuesto del código.

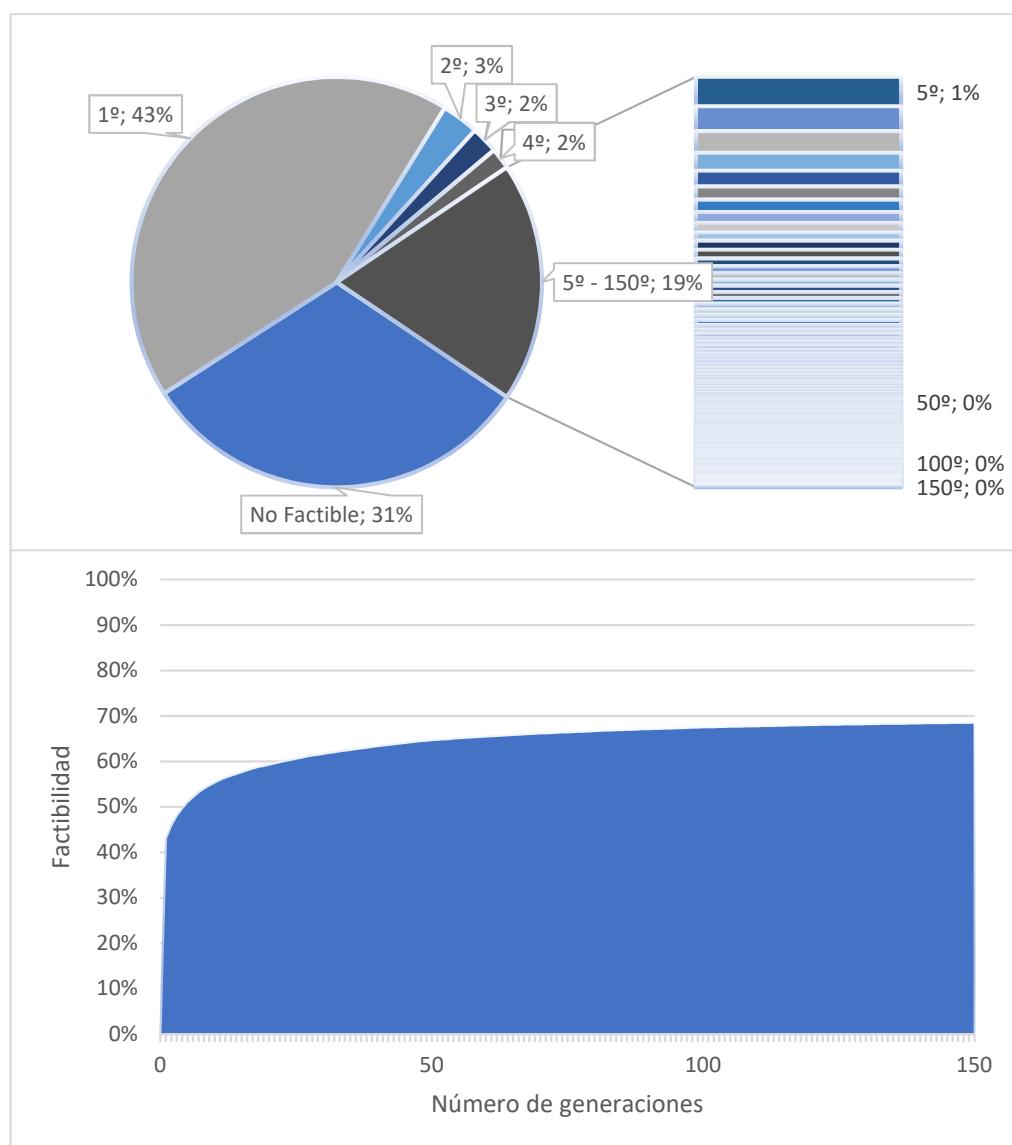


Ilustración 14. Gráficas de factibilidad en función del número de generaciones.

En la *Ilustración 14*, se observa a simple vista, que para la muestra la relación establecida entre la factibilidad y el número de generaciones parece seguir el comportamiento de una función logarítmica, con tan solo 5 generaciones ya se puede asegurar la factibilidad de más de la mitad de los escenarios. Para las pruebas se han tomado 50, 100 y 150 generaciones que dan lugar a un 65, 68 y 69 % de soluciones factibles respectivamente.

Si se aplica regresión logarítmica:

$$f(g) = 0,0514 * \ln(g) + 0,4379 \quad R^2 = 0,9869$$

Donde g es el número de generaciones y $f(g)$ la probabilidad de que sea una solución factible, con un coeficiente de determinación de casi 1, ergo es sin ninguna duda este tipo de función.

Continuando con el siguiente parámetro relevante, el código, en la *Tabla 4* se puede observar que sigue habiendo cierta dependencia al número de generaciones ya que, a excepción de Gerona, los expertos han optado por no emplear más de 50 generaciones, lo que para la función previa supone una probabilidad de éxito del 64%, siendo muy superior la observada tanto para Barcelona como para su costa.

Tabla 4. Factibilidad desglosada por código.

Código	Núm. casos	Med. Generaciones	Porcentaje factible
Barcelona	25758	50	0,9245
Gerona	26001	134	0,4395
costa	26244	50	0,8201
interior	30375	50	0,5778
TOTAL	108378	70	0,6857

Se puede comprender la baja tasa de factibilidad en la zona de interior, pues como se podía observar en la *Ilustración 13* son casos con tiempos de desplazamiento muy elevados que pueden resultar en el incumplimiento de los horarios para traslado y recogida del paciente. Esta afirmación queda contrastada al observar la tendencia de que a mayor tiempo de traslados mayor es la probabilidad de que la solución al problema sea no factible, siendo directamente imposible de realizar para traslados superiores a 300 minutos tal y como se puede comprobar en la *Ilustración 15*.

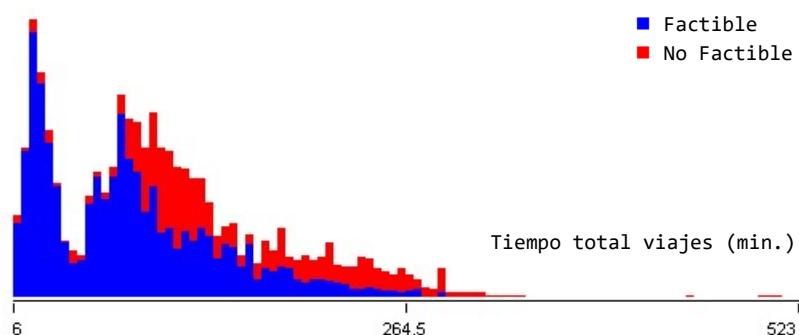


Ilustración 15. Relación entre tiempo total viajes y factibilidad.

Sin embargo, esto no se cumple para Gerona, que cuenta con una factibilidad muy por debajo del porcentaje predicho para toda la muestra y con tiempos de traslado relativamente cortos. Por ello la única hipótesis posible es que se deba a la gran cantidad de servicios a realizar. Al existir tantos servicios, queda en manos del experto considerar el número idóneo de dotaciones a emplear. Haciendo uso de un árbol J48 se puede observar un comportamiento similar al resto de la muestra hasta 70 servicios, pasado ese punto parece que el número de generaciones cobra menos relevancia y lo que asegura la factibilidad es el número de individuos, no pudiendo resolver el problema para aquéllos con menos de 150.

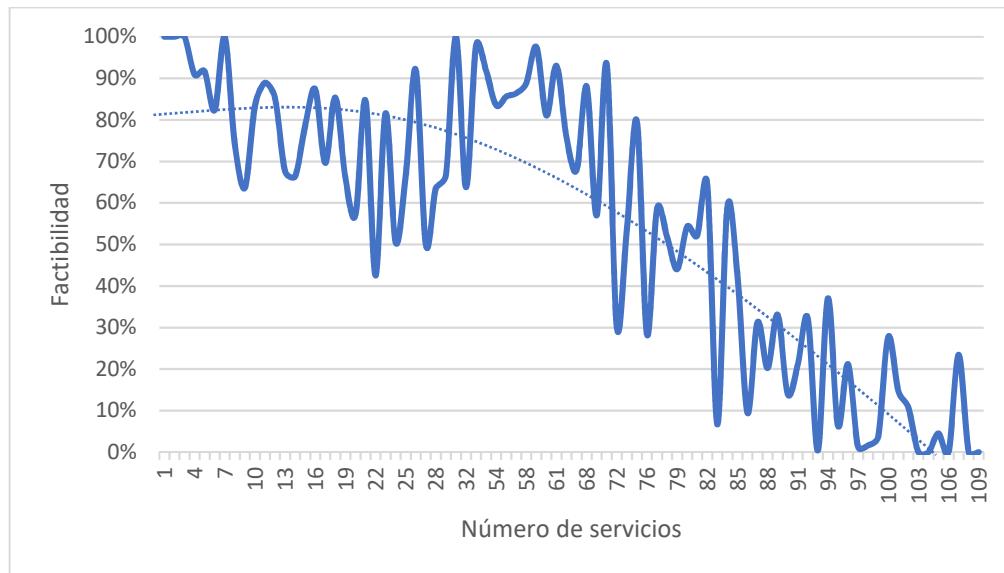


Ilustración 16. Factibilidad en función del número de servicios

Dando un rápido vistazo a la *Ilustración 16* lo primero que se observa es una fuerte tendencia lineal o polinómica, en la que, a mayor número de servicios menor factibilidad. Teniendo en cuenta los resultados del algoritmo J48 se puede contrastar un funcionamiento más ideal previo a la marca de 70 servicios con una media de factibilidad del 76%, y pasado ese punto la factibilidad cae en picado a una media de 25%, algo que es posiblemente resultado de no haber aumentado el número de individuos con respecto a la complejidad del caso a tratar.

5.2. Sostenibilidad

Una vez cumplidos los requisitos de calidad, el siguiente aspecto más importante es el de la sostenibilidad. Siempre que se cumplan los mínimos legales de calidad o aquéllos impuestos por la empresa, es deseable buscar un mejor aprovechamiento de las ambulancias para reducir tanto su huella de carbono como su impacto económico. Los indicadores serían:

- penalizacionTiempo
- usoAmbulancias
- tiempoTotalViaje
- pacientesViaje.

Como se cuentan con múltiples columnas, para simplificar el análisis se comienza por aplicar un PCA (Principal Component Analysis) para detectar dependencias y datos replicados, como es el caso del porcentaje de uso de ambulancias y el de tiempo total viaje, y siempre va a ser más útil el valor porcentual que el absoluto.

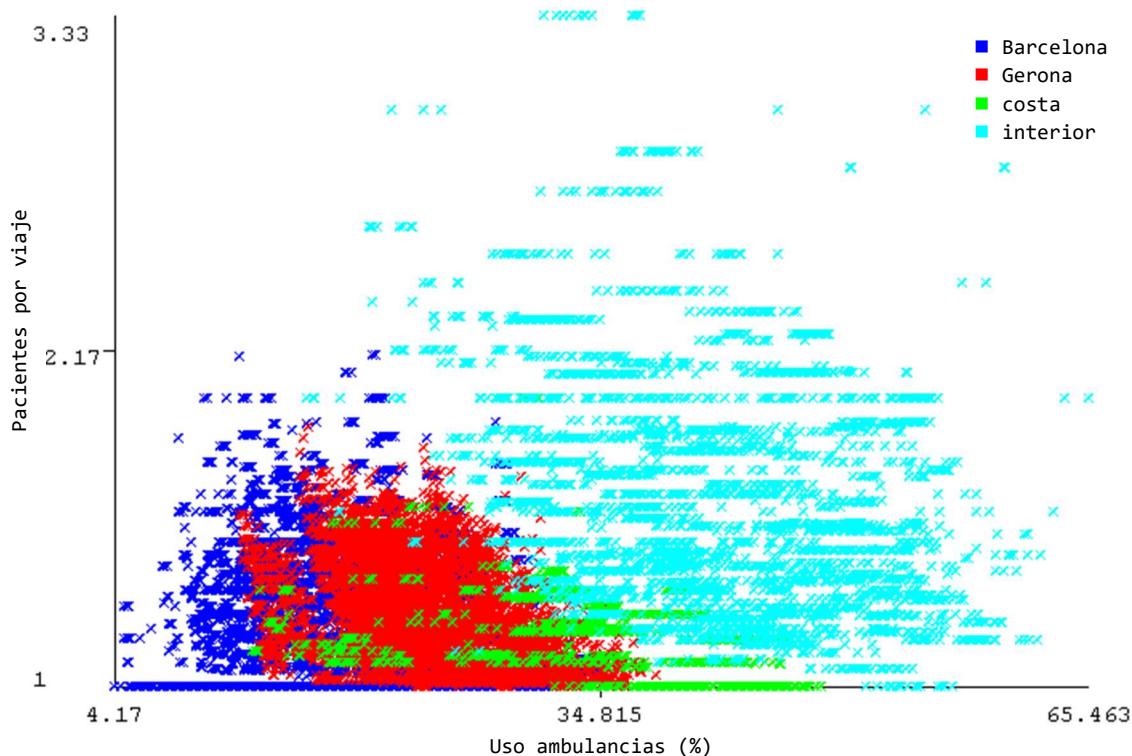


Ilustración 17. Relación entre el uso de ambulancias y los pacientes por viaje.

En la *Ilustración 17* se puede comprobar nuevamente la dependencia de la zona de actuación. Para esta gráfica se han omitido las soluciones no factibles, que representan un 30% de la muestra, pues no tendría sentido una solución muy sostenible pero irrealizable. Como se puede ver, el aprovechamiento de las ambulancias es mayor en el interior, en zonas rurales los tiempos de viaje son lógicamente mayores, y dado que por lo general los horarios de las dotaciones de las ambulancias son similares tanto en el campo como en la ciudad, el uso de las ambulancias suele resultar menor en las urbes.

Tabla 5. Media de pacientes por viaje y zona.

Código	1 paciente	(1,2] pacientes	(2,3] pacientes	(3,4] pacientes
Barcelona	7534	16267	13	0
Gerona	1735	9692	0	0
costa	13567	7957	0	0
interior	422	12558	4535	37

En cuanto a la capacidad para compartir ambulancia, desglosado en la *Tabla 5*, se puede observar que depende en parte del estado del paciente, ya que una camilla ocupa más que una silla de ruedas o un asiento. Como se puede comprobar, se dan unas pocas combinaciones de hasta 3 pacientes en Barcelona y solo se llegan a aprovechar realmente a máxima capacidad en la zona de interior en la que escasamente se producen traslados únicos, algo que no ocurre en el caso de la costa.

Como se ha detallado, nuevamente la zona de actuación supone un factor determinante en el resultado y quizás podría ser interesante ampliar el rango de actuación, es decir modificar los límites de las zonas designadas, en especial de las ciudades, englobando algún que otro núcleo poblacional próximo.

5.3. Resultados

En resumen, dadas las características del algoritmo hay un peso excesivo por parte de los datos que definen el caso, por lo que la solución no varía demasiado al manipular la mayoría de parámetros del algoritmo, más allá de ajustar aquellos con dependencia directa como son el número de generaciones y el de individuos. Aun así, del análisis previo se pueden extraer las siguientes recomendaciones a la hora de usar NURA:

- Tras definir el caso, realizar una primera ejecución de prueba con no más de 20 generaciones, como norma general pasado ese umbral la probabilidad de que sea factible es inferior al 10%. Verificar posible error humano, como horarios imposibles o transporte inapropiado (p. ej.: un paciente necesita ser trasladado en camilla y no hay ambulancias con esa configuración).
- La factibilidad en zonas rurales y amplias es menor, las rutas y distancias en una ciudad son más consistentes, mientras que, en el campo, al existir un menor número de vías, las decisiones del algoritmo son mucho más relevantes. Esto se traduce en que el campo responde mejor a un mayor número de generaciones y la ciudad a mayor número de individuos. El número de individuos debe aumentar de forma directamente proporcional al número de servicios.
- Si se desean realizar tests rápidamente para verificar una posible selección de dotaciones en un caso con un número de servicios considerable, se puede optar por aumentar el tamaño de torneo y de reemplazo, que, da como resultado muestras que convergen algo más deprisa, sobre un 10%. La solución estará muy lejos de ser ideal, pero como su tiempo de ejecución será algo más reducido se podrá verificar con antelación su idoneidad.
- Lograr una solución sostenible es complicado y recae principalmente en la pericia del experto a la hora de seleccionar la mejor combinación de dotaciones. NURA puede buscar la mejor ruta y las combinaciones de traslados para recorrer la mínima distancia, sin embargo, la tipología de ambulancias disponibles y los horarios de trabajo de las mismas son factores muy relevantes que le son externos.

Como posible mejora en NURA, se propone establecer un sistema que verifique que para todos los tipos de pacientes existe al menos una ambulancia compatible con sus necesidades de traslado. Es algo común que le puede suceder a un usuario no-experto, pero siempre se puede cometer un error y es preferible que se avise de este error antes de ejecutar todo el algoritmo genético para obtener indudablemente una solución no factible.

6. Conclusiones

En primer lugar, considero de gran importancia todo lo relacionado con el apartado de la inteligencia artificial de cara a un futuro profesional, con independencia de su rama, siendo hoy en día como son tan relevantes la visión por computador, el procesamiento de lenguaje natural y el aprendizaje automático. Este proyecto me ha servido para profundizar en esta última disciplina, pues la herramienta de data mining sirve precisamente para facilitar su uso.

He podido ser consciente de la gran cantidad de soluciones matemáticas que existen actualmente en torno al machine learning, y de su utilidad e importancia estratégica en la inteligencia de negocio de cualquier empresa, dada su capacidad para aportar conocimiento que no se puede conocer bajo una observación más simple de los datos que se disponga.

También hay que destacar la posibilidad de toma de contacto con un algoritmo logístico en producción, y lo más importante, la oportunidad de comprenderlo en profundidad, pues debía ser capaz de entender su funcionamiento si quería analizarlo. Los algoritmos genéticos son una de las soluciones técnicas preferentes en temas de planificación y se está extendiendo su uso a campos como el diseño industrial o la prescripción de medicamentos, entre otros. Se convierten así en una tecnológica básica a tener en cuenta por cualquiera conocedor de la materia.

Evidentemente, con el desarrollo de la aplicación de gestión de correcciones he podido poner en práctica los conocimientos adquiridos durante mis estudios, teniendo en cuenta la complejidad de la misma por su alta concurrencia y la necesidad de desarrollar una interfaz amigable y fácil de utilizar. Cabe destacar la relevancia de la implementación de software con arquitecturas concurrentes capaces de aprovechar mejor los recursos que pone a nuestra disposición el hardware contemporáneo. El diseño multihilo suele ser ignorado para aplicaciones sencillas, en las que una mejora del 26% no supone demasiado, y no logramos apreciar su utilidad hasta que no nos encontramos ante tiempos de ejecución elevados en los que cobra vital importancia la escalabilidad.

Para terminar, he sufrido los efectos adversos del procesamiento manual de datos, ya que siempre que sea posible es preferible usar un ETL automático para prevenir los inevitables errores humanos. Asimismo, cualquier sistema en origen que hubiese transformado todos los datos a catalán o a castellano habría prevenido gran parte del problema de redundancia en la base de datos y el posterior consumo de servicios de geolocalización.

Bibliografía

- Fogue, M., Sanguesa, J. A., Naranjo, F., Gallardo, J., Garrido, P., & Martinez, F. J. (1 de Noviembre de 2016). Non-emergency patient transport services planning through genetic algorithms. *Expert Systems with Applications*, 61, 262-271. doi:10.1016/j.eswa.2016.05.028
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. United States: Addison-Wesley.
- Melián, B., Moreno, J. A., & Moreno, J. M. (Agosto de 2009). Algoritmos Genéticos. Una visión práctica. *Números*, 71, 29-47.
- Melián, B., Moreno, J. A., & Moreno, J. M. (Agosto de 2009). Introducción a la Computación Evolutiva. *Números*, 71, 21-27.
- Verdugo, A. B., & López, M. I. (2020). El transporte sanitario no urgente (TSNU): el lado más humano del transporte sanitario. *ZonaTES*, 9(2).
- Wikipedia. (26 de Marzo de 2020). *Algoritmo genético*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico

Anexos

Anexo I. Script para dar soporte a NuraTools en la BD de NURA

```
-- Se crean las nuevas tablas
CREATE TABLE `tsc_pre`.`geolocalizadores` (
  `ID` INT(11) NOT NULL AUTO_INCREMENT,
  `NOMBRE` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`ID`));

-- Alteraciones en la tabla coordenadas, que ninguna coordenada este
-- validada y que si se alteran las FK se seten a null para la referencia a
-- otros geolocalizadores
ALTER TABLE `tsc_pre`.`coordenadas`
ADD COLUMN `COORD_TIPO` VARCHAR(30) NOT NULL DEFAULT 'CERCALIA'
  AFTER `LONGITUD`,
ADD COLUMN `VALIDADO` TINYINT(1) NOT NULL DEFAULT 0
  AFTER `COORD_TIPO`;

-- Similar para duracion viajes
ALTER TABLE `tsc_pre`.`duracionviajes`
ADD COLUMN `DURAC_TIPO` VARCHAR(30) NOT NULL DEFAULT 'CERCALIA'
  AFTER `FECHA_INSERCIÓN`,
ADD COLUMN `VALIDADO` TINYINT(1) NOT NULL DEFAULT 0
  AFTER `DURAC_TIPO`;

-- Por defecto, el tipo de proveedor es de Cercalia, se definen las PK
-- como INT (32) 4 billones de coordenadas y de rutas posibles
CREATE TABLE `tsc_pre`.`coordenadasalt` (
  `ID` INT(32) NOT NULL AUTO_INCREMENT,
  `TIPOVIA` VARCHAR(20) NOT NULL,
  `VIA` VARCHAR(50) NOT NULL,
  `NUMERO` INT(11) NOT NULL,
  `POBLACION` VARCHAR(50) NOT NULL,
  `PROVINCIA` VARCHAR(30) NOT NULL,
  `LATITUD` DECIMAL(10,6) NOT NULL,
  `LONGITUD` DECIMAL(10,6) NOT NULL,
  `COORDALT_TIPO` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`ID`),
  FOREIGN KEY (`TIPOVIA`, `VIA`, `NUMERO`, `POBLACION`, `PROVINCIA`)
  REFERENCES
`tsc_pre`.`coordenadas`(`TIPOVIA`, `VIA`, `NUMERO`, `POBLACION`, `PROVINCIA`)
  ON DELETE CASCADE ON UPDATE CASCADE
);

-- Trigger de update para eliminar las alternativas al validar la coordenada
DELIMITER $$

CREATE TRIGGER au_coordenadas AFTER UPDATE ON coordenadas FOR EACH ROW
  IF NEW.VALIDADO <> OLD.VALIDADO AND NEW.VALIDADO = 1 THEN
    DELETE FROM coordenadasalt
    WHERE TIPOVIA = OLD.TIPOVIA AND VIA = OLD.VIA AND
      NUMERO = OLD.NUMERO AND POBLACION = OLD.POBLACION AND
      PROVINCIA = OLD.PROVINCIA;
  END IF; $$
```

```
-- Similar para las alternativas de duracion viajes
CREATE TABLE `tsc_pre`.`duracionviajesalt` (
  `ID` INT(32) NOT NULL AUTO_INCREMENT,
  `LAT_ORIGEN` DECIMAL(10,6) NOT NULL,
  `LON_ORIGEN` DECIMAL(10,6) NOT NULL,
  `LAT_DESTINO` DECIMAL(10,6) NOT NULL,
  `LON_DESTINO` DECIMAL(10,6) NOT NULL,
  `DISTANCIA` DOUBLE NOT NULL,
  `DURACION` INT(11) NOT NULL,
  `DURACALT_TIPO` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`ID`),
  FOREIGN KEY (`LAT_ORIGEN`, `LON_ORIGEN`, `LAT_DESTINO`, `LON_DESTINO`)
  REFERENCES
`tsc_pre`.`duracionviajes`(`LAT_ORIGEN`, `LON_ORIGEN`, `LAT_DESTINO`, `LON_DESTINO`)
  ON DELETE CASCADE ON UPDATE CASCADE,
);

```

NURA Tools

Manual de usuario

Versión 1.0



INSTITUTO NACIONAL
DE ESTADÍSTICA Y GEOGRAFÍA

15 de julio de 2020

Tabla de contenido

1.- Objetivo del sistema.....	4
1.1.- Objeto y alcance.....	4
1.2.- Funcionalidad	4
2.- Mapa del Sistema	4
2.1.- Modelo Lógico.....	4
2.2.- Navegación	4
3.- Descripción del sistema.....	5
3.1.- Ajustes	6
3.2.- Direcciones	7
3.3.- Distancias	10
3.4.- Usuarios.....	12
3.5.- Zonas y bases	13
4.- Alertas del sistema	14
5.- FAQ	16



Tabla de ilustraciones

Ilustración 1. Ventanas de navegación de la aplicación	4
Ilustración 2. Ventana de inicio de la aplicación	5
Ilustración 3. Barra de herramientas.....	5
Ilustración 4. Estados de la automatización.....	6
Ilustración 5. Botones de navegación	6
Ilustración 6. Pestaña de ajustes.....	6
Ilustración 7. Herramientas de contexto de la pestaña de direcciones.....	7
Ilustración 8. Tabla de direcciones	7
Ilustración 9. Ventana emergente para elementos sin proveer.....	8
Ilustración 10. Ventana de edición de coordenadas (Cercalia).....	9
Ilustración 11. Ventana de edición de coordenadas (Manual)	9
Ilustración 12. Confirmación de cambios.....	10
Ilustración 13. Tabla de distancias	10
Ilustración 14. Ventana de edición de rutas.....	11
Ilustración 15. Ventana de confirmación de cambios	11
Ilustración 16. Pestaña de usuarios.....	12
Ilustración 17. Menú desplegable	12
Ilustración 18. Editando usuarios.....	13
Ilustración 19. Tabla de bases	13
Ilustración 20. Existe otra instancia de esta aplicación.....	14
Ilustración 21. Excepción SQL (inicio).....	14
Ilustración 22. Excepción SQL.....	14
Ilustración 23. Error de conexión remota	15
Ilustración 24. Error grave de interpretación.....	15
Ilustración 25. Google no tiene saldo suficiente.....	15



1.- Objetivo del sistema

1.1.- Objeto y alcance

Este documento contiene información relativa a la funcionalidad, navegación y uso de la aplicación NURA Tools.

1.2.- Funcionalidad

El programa tiene como objetivo la validación automática de direcciones y distancias necesarias para organizar traslados sanitarios de carácter no urgente. Para ello hace uso de dos proveedores online de servicios de geolocalización, Cercalia y Google.

Por otra parte, también sirve para administrar a los usuarios, zonas de actuación y bases.

2.- Mapa del Sistema

2.1.- Modelo Lógico

El sistema se compone de 4 módulos:

- **Direcciones:** Permite la validación de direcciones, asignando las geocoordenadas idóneas con ayuda de un mapa y de los proveedores.
- **Distancias:** Permite la validación de distancias, ya que dadas dos geocoordenadas se debe tener la distancia en metros y el tiempo necesario en cubrir dicha distancia.
- **Usuarios:** Permite la administración de los usuarios que acceden a la aplicación principal, adjudicando las zonas en las que éstos pueden operar. Contiene las herramientas básicas de edición (crear, borrar, duplicar y editar).
- **Zonas y Bases:** Permite administrar las zonas donde se tienen los servicios y las bases que dichas zonas contienen.

2.2.- Navegación

La aplicación cuenta con 4 ventanas principales, una por cada módulo definido previamente, tal como se observa en la *Ilustración 1*. Ventanas de navegación de la aplicación. Tanto el apartado de direcciones como el de distancias, cuentan con ventanas emergentes para editar sus valores.

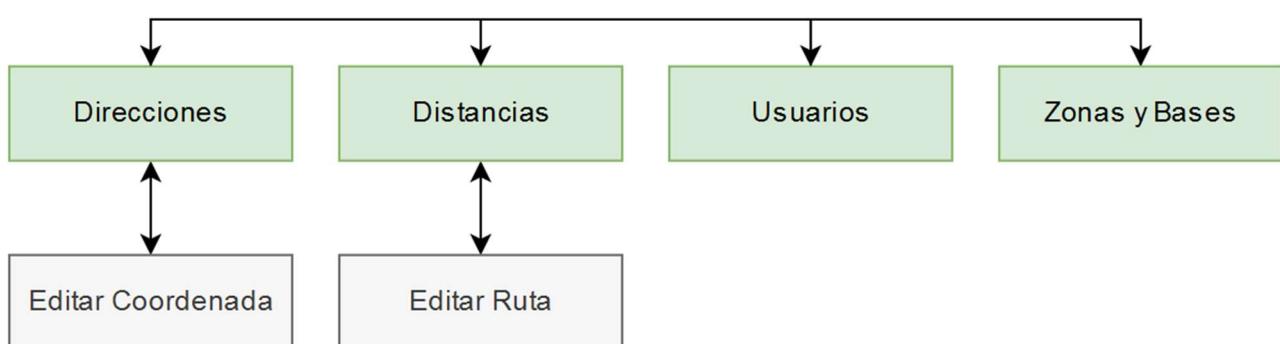


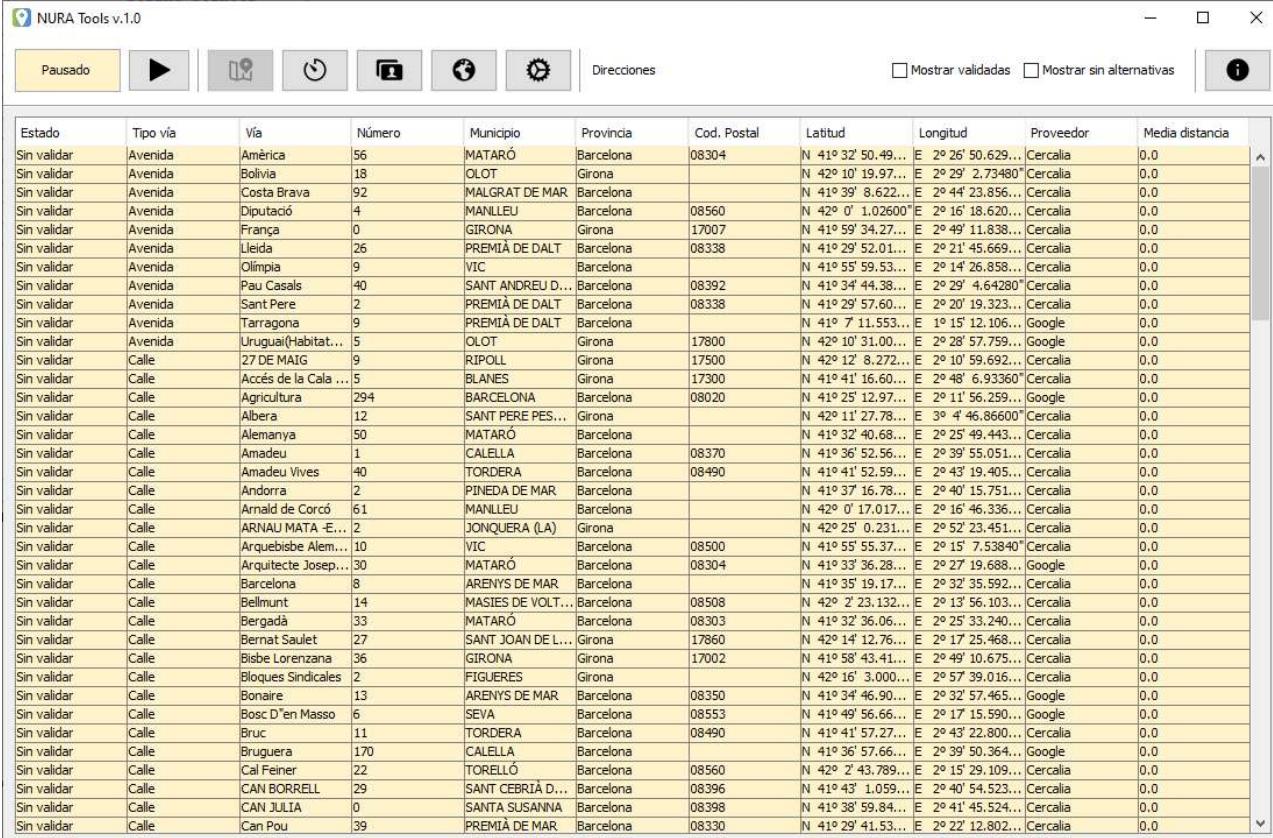
Ilustración 1. Ventanas de navegación de la aplicación



3.- Descripción del sistema

NuraTools es una aplicación de gestión, por lo que al comienzo de la misma es necesario recopilar los datos almacenados en el servidor.

Una vez arrancada, tal como se puede observar en la *Ilustración 2*. Ventana de inicio de la aplicación, la aplicación está compartimentada en dos secciones. La superior, de fondo blanco, contiene la barra de herramientas, mientras que la de mayor tamaño, de fondo gris, representa el área de trabajo.



Direcciones										
<input type="checkbox"/> Mostrar validadas <input type="checkbox"/> Mostrar sin alternativas										
Estado	Tipo vía	Vía	Número	Municipio	Provincia	Cod. Postal	Latitud	Longitud	Proveedor	Media distancia
Sin validar	Avenida	América	56	MATARÓ	Barcelona	08304	N 41° 32' 50.49...	E 2° 26' 50.629...	Cercalia	0,0
Sin validar	Avenida	Bolivia	18	OLOT	Girona	08036	N 42° 10' 19.97...	E 2° 29' 2.73480...	Cercalia	0,0
Sin validar	Avenida	Costa Brava	92	MALGRAT DE MAR	Barcelona	08840	N 41° 39' 8.622...	E 2° 44' 23.856...	Cercalia	0,0
Sin validar	Avenida	Deputació	4	MANlleu	Barcelona	08560	N 42° 0' 1.02600...	E 2° 16' 18.620...	Cercalia	0,0
Sin validar	Avenida	França	0	GIRONA	Girona	17007	N 41° 59' 34.27...	E 2° 49' 11.838...	Cercalia	0,0
Sin validar	Avenida	Lleida	26	PREMIÀ DE DALT	Barcelona	08338	N 41° 29' 52.01...	E 2° 21' 45.669...	Cercalia	0,0
Sin validar	Avenida	Olimpia	9	VIC	Barcelona	08034	N 41° 55' 59.53...	E 2° 14' 26.858...	Cercalia	0,0
Sin validar	Avenida	Pau Casals	40	SANT ANDREU D...	Barcelona	08392	N 41° 34' 44.38...	E 2° 29' 4.64280...	Cercalia	0,0
Sin validar	Avenida	Sant Pere	2	PREMIÀ DE DALT	Barcelona	08338	N 41° 29' 57.60...	E 2° 20' 19.323...	Cercalia	0,0
Sin validar	Avenida	Tarragona	9	PREMIÀ DE DALT	Barcelona	08338	N 41° 7' 11.553...	E 1° 15' 12.106...	Google	0,0
Sin validar	Avenida	Uruguay(+habitac...)	5	OLOT	Girona	17800	N 42° 10' 31.00...	E 2° 28' 57.759...	Google	0,0
Sin validar	Calle	27 DE MAIG	9	RIPOLL	Girona	17500	N 42° 12' 8.272...	E 2° 10' 59.692...	Cercalia	0,0
Sin validar	Calle	Accés de la Cala ...	5	BLANES	Girona	17300	N 41° 41' 16.60...	E 2° 48' 6.93360...	Cercalia	0,0
Sin validar	Calle	Agricultura	294	BARCELONA	Barcelona	08020	N 41° 25' 12.97...	E 2° 11' 56.259...	Google	0,0
Sin validar	Calle	Albera	12	SANT PERE PES...	Girona	17300	N 42° 11' 27.78...	E 3° 4' 46.86600...	Cercalia	0,0
Sin validar	Calle	Alemanyia	50	MATARÓ	Barcelona	08304	N 41° 32' 40.68...	E 2° 25' 49.443...	Cercalia	0,0
Sin validar	Calle	Amadeu	1	CALELLA	Barcelona	08370	N 41° 36' 52.56...	E 2° 39' 55.051...	Cercalia	0,0
Sin validar	Calle	Amadeu Vives	40	TORDERA	Barcelona	08490	N 41° 41' 52.59...	E 2° 43' 19.405...	Cercalia	0,0
Sin validar	Calle	Andorra	2	PINEDA DE MAR	Barcelona	08508	N 41° 37' 16.78...	E 2° 40' 15.751...	Cercalia	0,0
Sin validar	Calle	Arnald de Corcò	61	MANlleu	Barcelona	08560	N 42° 0' 17.017...	E 2° 16' 46.336...	Cercalia	0,0
Sin validar	Calle	ARNAU MATA -E...	2	JONQUERA (LA)	Girona	17860	N 42° 25' 0.231...	E 2° 52' 23.451...	Cercalia	0,0
Sin validar	Calle	Arquebisbe Àlem...	10	VIC	Barcelona	08500	N 41° 55' 55.37...	E 2° 15' 7.53840...	Cercalia	0,0
Sin validar	Calle	Arquitecte Josep...	30	MATARÓ	Barcelona	08304	N 41° 33' 36.28...	E 2° 27' 19.688...	Google	0,0
Sin validar	Calle	Barcelona	8	ARENYS DE MAR	Barcelona	08350	N 41° 35' 19.17...	E 2° 32' 35.592...	Cercalia	0,0
Sin validar	Calle	Bellmunt	14	MASIES DE VOLT...	Barcelona	08508	N 42° 2' 23.132...	E 2° 13' 56.103...	Cercalia	0,0
Sin validar	Calle	Bergadà	33	MATARÓ	Barcelona	08303	N 41° 32' 36.06...	E 2° 25' 33.240...	Cercalia	0,0
Sin validar	Calle	Bernat Saulet	27	SANT JOAN DE L...	Girona	17860	N 42° 14' 12.76...	E 2° 17' 25.468...	Cercalia	0,0
Sin validar	Calle	Bisbe Lorenzana	36	GIRONA	Girona	17002	N 41° 58' 43.41...	E 2° 49' 10.675...	Cercalia	0,0
Sin validar	Calle	Blocs Sindicals	2	FIGUERES	Girona	17300	N 42° 16' 3.000...	E 2° 57' 39.016...	Cercalia	0,0
Sin validar	Calle	Bonaire	13	ARENYS DE MAR	Barcelona	08350	N 41° 34' 46.90...	E 2° 32' 57.465...	Google	0,0
Sin validar	Calle	Bosc D'en Masso	6	SEVA	Barcelona	08553	N 41° 49' 56.66...	E 2° 17' 15.590...	Google	0,0
Sin validar	Calle	Bruc	11	TORDERA	Barcelona	08490	N 41° 41' 57.27...	E 2° 43' 22.800...	Cercalia	0,0
Sin validar	Calle	Bruguera	170	CALELLA	Barcelona	08370	N 41° 36' 57.66...	E 2° 39' 50.364...	Google	0,0
Sin validar	Calle	Cal Feiner	22	TORELLÓ	Barcelona	08560	N 42° 2' 43.789...	E 2° 15' 29.109...	Cercalia	0,0
Sin validar	Calle	CAN BORRELL	29	SANT CEBRIÀ D...	Barcelona	08396	N 41° 43' 1.059...	E 2° 40' 54.523...	Cercalia	0,0
Sin validar	Calle	CAN JULIA	0	SANTA SUSANNA	Barcelona	08398	N 41° 38' 59.84...	E 2° 41' 45.524...	Cercalia	0,0
Sin validar	Calle	Can Pou	39	PREMIÀ DE MAR	Barcelona	08330	N 41° 29' 41.53...	E 2° 22' 12.802...	Cercalia	0,0

Ilustración 2. Ventana de inicio de la aplicación

La barra de herramientas se reparte a su vez en 4 subapartados: automatización (1), navegación (2), herramientas de contexto (3) e información (4).



Ilustración 3. Barra de herramientas

1. Un botón para gestionar la automatización, activando y desactivando la validación automática, y un indicador de estado. Los estados se detallan en la *Ilustración 4*.



Pausado: A la espera de que el usuario active la automatización

En ejecución: En ejecución y mostrando el tiempo de respuesta de los proveedores, o lo que es lo mismo, lo que se tarda en generar alternativas.

Finalizado: El proceso está parado y no puede ser reactivado. Este estado se alcanza al acabar con todas las direcciones y distancias o cuando se produzca algún error grave.

Ilustración 4. Estados de la automatización

2. Botones para la navegación: configuración de ajustes, tabla de direcciones, tabla de rutas, usuarios y zonas, y bases. Se detallan más adelante.



Ilustración 5. Botones de navegación

3. Pestaña activa y herramientas de contexto. Estas últimas son opciones adicionales para cada uno de los apartados previos (ver *Ilustración 7*).
4. Botón con información relativa al programa, autoría, descripción y fuentes.

3.1.- Ajustes

El botón de ajustes de configuración de la automatización muestra la siguiente pestaña, con dos variables y una checkbox

ATENCIÓN: Ante políticas menos restrictivas se recomienda una revalidación de todo el contenido.

Distancia máx. coordenadas	<input type="text" value="0.2"/> Km
Diferencia máx. relat. distan.	<input type="text" value="30"/> %
<input type="checkbox"/> Comenzar autovalidación al inicio de la aplicación	

Ilustración 6. Pestaña de ajustes

La primera variable hace referencia a la distancia máxima, en kilómetros, permitida con respecto a la coordenada actual. Tal como aparece en la *Ilustración 6*, al generar alternativas si la distancia a la establecida de alguna de ellas es superior a 200 metros será necesaria la validación por parte del experto.

La segunda variable es la diferencia máxima relativa a distancias. De carácter porcentual, autovalida si hay una diferencia menor a la establecida entre dos distancias. Para este caso (30%), si tenemos un viaje con una duración de 100 minutos, la aplicación no consultará al usuario a no ser que uno de los proveedores diga que para cubrir esa distancia son necesarios 130 minutos o más.



La checkbox sirve para que, al iniciar, la aplicación comience o no a autovalidar.

3.2.- Direcciones

Es la pantalla que nos recibe por defecto, cuenta con dos checkboxes como herramientas de contexto, que nos permiten mostrar y ocultar las direcciones validadas y aquéllas sin alternativas

Mostrar validadas Mostrar sin alternativas

Ilustración 7. Herramientas de contexto de la pestaña de direcciones

Si marcamos todas las opciones podríamos ver algo similar a la Ilustración 8, es decir, una tabla que puede ser ordenada por cualquiera de sus columnas, haciendo click izquierdo en sus cabeceras.

Estado	Tipo vía	Vía	Número	Municipio	Provincia	Cod. Postal	Latitud	Longitud	Proveedor	Media distancia
Sin cargar	Barcelona	Transversal	47	MATARÓ	Barcelona					0,0
Sin cargar	Barcelona	Eduard Maristany	21	SANT ADRIÀ DE ...	Barcelona					0,0
Sin cargar	Barcelona	Diagonal	16	BARCELONA	Barcelona					0,0
Sin cargar	Barcelona	Astronàutica	9	SANT ADRIÀ DE ...	Barcelona					0,0
Sin cargar	Barcelona	Indústria	84	MALGRAT DE MAR	Barcelona					0,0
Sin cargar	Barcelona	Nacional 2	669	PINEDA DE MAR	Barcelona					0,0
Sin cargar	Barcelona	Colòmbia	25	MATARÓ	Barcelona					0,0
Sin cargar	Barcelona	Can Vehils	0	CABRILS	Barcelona					0,0
Sin resultados	Barcelona	Tres Tombs	16	MATARÓ	Barcelona					0,0
Sin resultados	Barcelona	Països Catalans	0	MATARÓ	Barcelona					0,0
Sin resultados	Barcelona	Hospital	31	MATARÓ	Barcelona					0,0
Sin alternativas	.	2	SALT	Girona	17190	N 41° 58' 26.97... E 2° 47' 33.302...			Cercalia	0,0
Sin alternativas	Alemania	41	Figuera		17600	N 42° 15' 40.99... E 29° 58' 44.860...			Cercalia	0,0
Sin alternativas		Antonio Gargallo	2	Teruel		44002	N 40° 19' 45.71... W 1° 5' 30.717...		Cercalia	0,0
Sin Validar		Can Moianès	2	BALENYÀ	Barcelona	08550	N 41° 49' 54.61... E 2° 14' 1.23000		Google	2,067
Sin Validar		COLLEGI	3	SANT MIQUEL D...	Girona		N 41° 58' 37.66... E 2° 48' 46.231...		Google	149,385
Validado		CTRÀ DE L'ESTA...	85	TORROELLA DE ...	Girona	17257	N 42° 2' 29.760...	E 3° 8' 3.42600	Cercalia	0,0
Validado		DISSEMINAT - M...	2	CASTELLÓ D'EM...	Girona	17466	N 42° 15' 29.72... E 3° 4' 27.98040		Cercalia	0,0
Validado		espana	6	zaragoza		50001	N 41° 39' 5.590...	W 0° 52' 50.67...	Cercalia	0,0
Validado		Gas	15	REUS	Tarragona	43204	N 41° 9' 8.47800	E 1° 6' 48.88440	Cercalia	0,0
Validado		Germans Sabat	141	GIRONA	Girona	17007	N 41° 59' 50.60...	E 2° 48' 10.782...	Cercalia	0,0
Validado		Girona per Girona	10	GIRONA	Girona	17003	N 41° 58' 18.48...	E 2° 49' 46.668...	Cercalia	0,0
Validado		Miquel Mas	48	CAPELLADES	Barcelona	08786	N 41° 31' 57.82...	E 1° 41' 0.65400	Cercalia	0,0
Validado		Nen de les Oques	15	REUS	Tarragona	43202	N 41° 9' 21.992...	E 1° 6' 5.90400	Cercalia	0,0
Validado		Sant celoni	6	Arenys de munt		08358	N 41° 36' 29.62...	E 2° 32' 28.219...	Cercalia	0,0
Validado	Avenida	América	11	MATARÓ	Barcelona	08304	N 41° 32' 51.64...	E 2° 26' 43.645...	Cercalia	0,0
Validado	Avenida	América	56	MATARÓ	Barcelona	08304	N 41° 32' 50.49...	E 2° 26' 50.629...	Cercalia	0,0
Validado	Avenida	Argentina	95	SANT ANDREU D...	Barcelona	08740	N 41° 26' 47.32...	E 1° 58' 38.701...	Cercalia	0,0
Validado	Avenida	ARRIATE	30	JONQUERA (LA)	Girona	17700	N 42° 25' 19.09...	E 2° 52' 6.55680	Cercalia	0,0
Validado	Avenida	ASSUMPTA	31	BESCANÓ	Girona	17162	N 41° 57' 50.58...	E 2° 44' 18.099...	Cercalia	0,0
Validado	Avenida	Astruc Ravaya	31	PALAMÓS	Girona	17230	N 41° 51' 8.582...	E 3° 7' 59.12400	Cercalia	0,0
Validado	Avenida	Ave María	25	ESCALA (L")	Girona	17130	N 42° 7' 23.476...	E 3° 8' 0.87000	Cercalia	0,0
Autovalidado	Calle	Badalona	10	ARGENTONA	Barcelona	08310	N 41° 33' 26.21...	E 2° 24' 11.700...	Google	0,0
Validado	Avenida	Balears	6	SANT FELIU DE ...	Girona	17220	N 41° 47' 21.21...	E 3° 1' 46.18560	Cercalia	0,0
Validado	Avenida	Balmes	74	IGUALADA	Barcelona	08700	N 41° 35' 11.72...	E 1° 36' 51.123...	Cercalia	0,0
Validado	Avenida	Barcelona	19	PIERA	Barcelona	08784	N 41° 31' 11.86...	E 1° 45' 31.269...	Cercalia	0,0
Validado	Avenida	Barcelona	51	VILAFRANCA DE...	Barcelona	08720	N 41° 20' 46.10...	E 1° 42' 10.022...	Cercalia	0,0

Ilustración 8. Tabla de direcciones

La tabla cuenta con 11 columnas:

- **Estado:** Una dirección puede contar con 6 estados:
 - **Sin cargar (rojo):** direcciones importadas no geolocalizadas.
 - **Sin resultados (rojo):** direcciones para las que no se ha logrado obtener ninguna coordenada. Requieren de la atención del usuario, pues los proveedores no han sido capaces de generar ningún resultado, por lo que no se puede tratar automáticamente.
 - **Sin alternativas (blanco):** la automatización las irá convirtiendo poco a poco en direcciones autovalidadas o sin validar. Si queremos que se valide inmediatamente



alguna de estas podemos hacer doble click sobre ella y nos aparecerá el desplegable de la *Ilustración 9*.

- **Sin validar (naranja):** tenemos aquéllas que cuentan con alternativas. Podemos ver que en la última columna cuentan con la distancia media entre dichas alternativas.
 - **Autovalidado (azul):** direcciones que no merecen nuestra atención, pues suponemos que son correctas.
 - **Validado (verde):** direcciones que han sido validadas por el usuario.
-
- **Tipo vía:** Tipo de la vía de la dirección, p.ej.: calle, avenida, plaza...
 - **Vía:** Nombre de la vía de la dirección.
 - **Número:** Número de portal de la dirección.
 - **Municipio:** Población de la dirección.
 - **Provincia:** Provincia de la dirección.
 - **Cód. Postal:** Código postal de la dirección. De especial importancia para mejorar la precisión de los proveedores
 - **Latitud:** Latitud de la coordenada asignada a la dirección.
 - **Longitud:** Longitud de la coordenada asignada a la dirección.
 - **Proveedor:** Organización que ha proporcionado la coordenada.
 - **Media distancia:** Solo cuenta con valor para las direcciones sin validar, siendo la media de las distancias entre la coordenada actual y sus alternativas

Si hacemos doble click en una dirección sin validar o sin resultados, nos aparecerá la ventana emergente de la *Ilustración 9*.

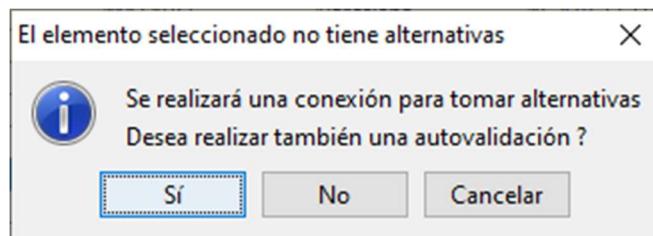


Ilustración 9. Ventana emergente para elementos sin proveer

Nos consulta acerca de la cargar alternativas y aquí podemos elegir si autovalidar o no. Esto nos permite no tener en cuenta los criterios de configuración.

En caso de que seleccionemos 'Sí', y no pueda autovalidar, o 'No', se desplegará la misma ventana de edición (*Ilustraciones 9 y 10*), que si hubiésemos seleccionado una dirección sin validar o hayamos seleccionado una ya (auto)validada.

Para el caso de las direcciones validadas se puede notar un pequeño retraso a la hora de abrir la ventana de edición, dado que también toma alternativas de nuevo, en este caso sin preguntar, para facilitar la tarea del usuario que esté revisando alguna dirección autovalidada que se considere errónea.



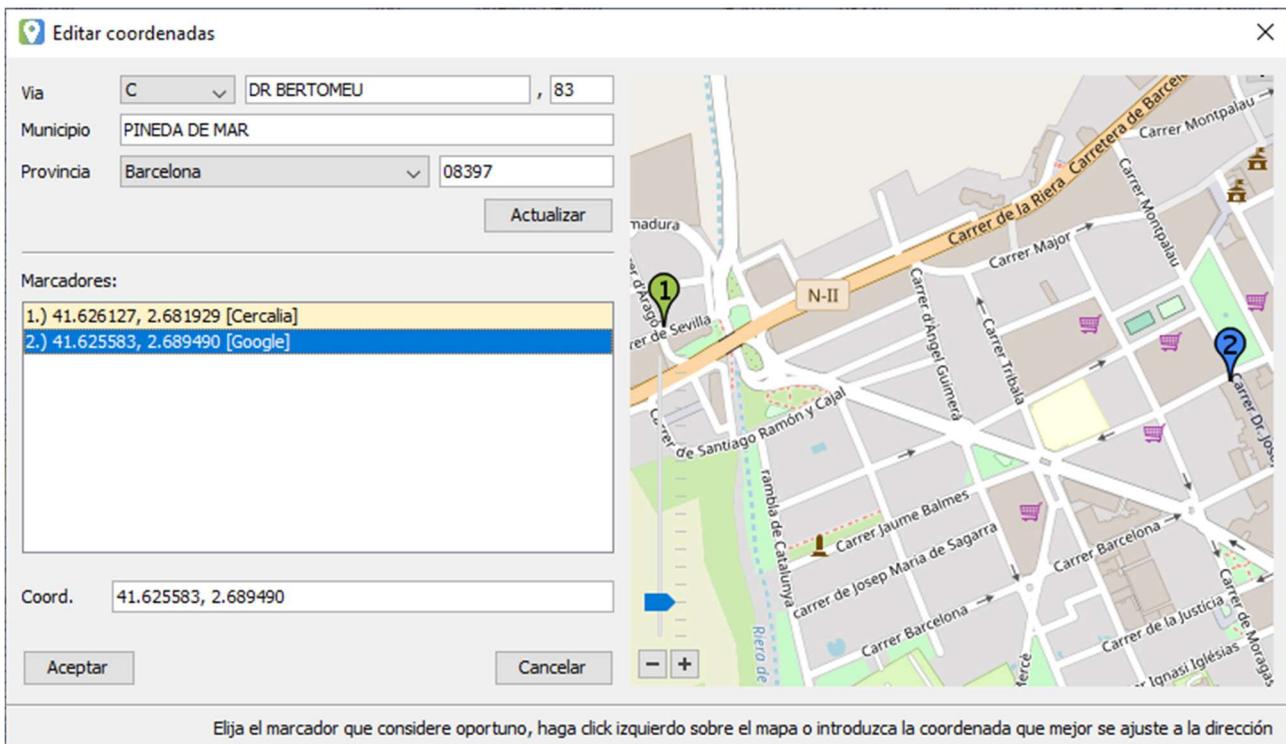


Ilustración 10. Ventana de edición de coordenadas (Cercalia)

Para este caso tenemos 2 opciones, el marcador de la izquierda es el original (porque es el primero de la lista y está destacado en naranja), el azul de la derecha es la opción de Google. Podemos seleccionar cualquiera de los 2, o añadir manualmente uno, haciendo click izquierdo sobre el mapa o introduciendo directamente la latitud y longitud. En la *Ilustración 11*, aparecerá el nombre de usuario que lo provea.

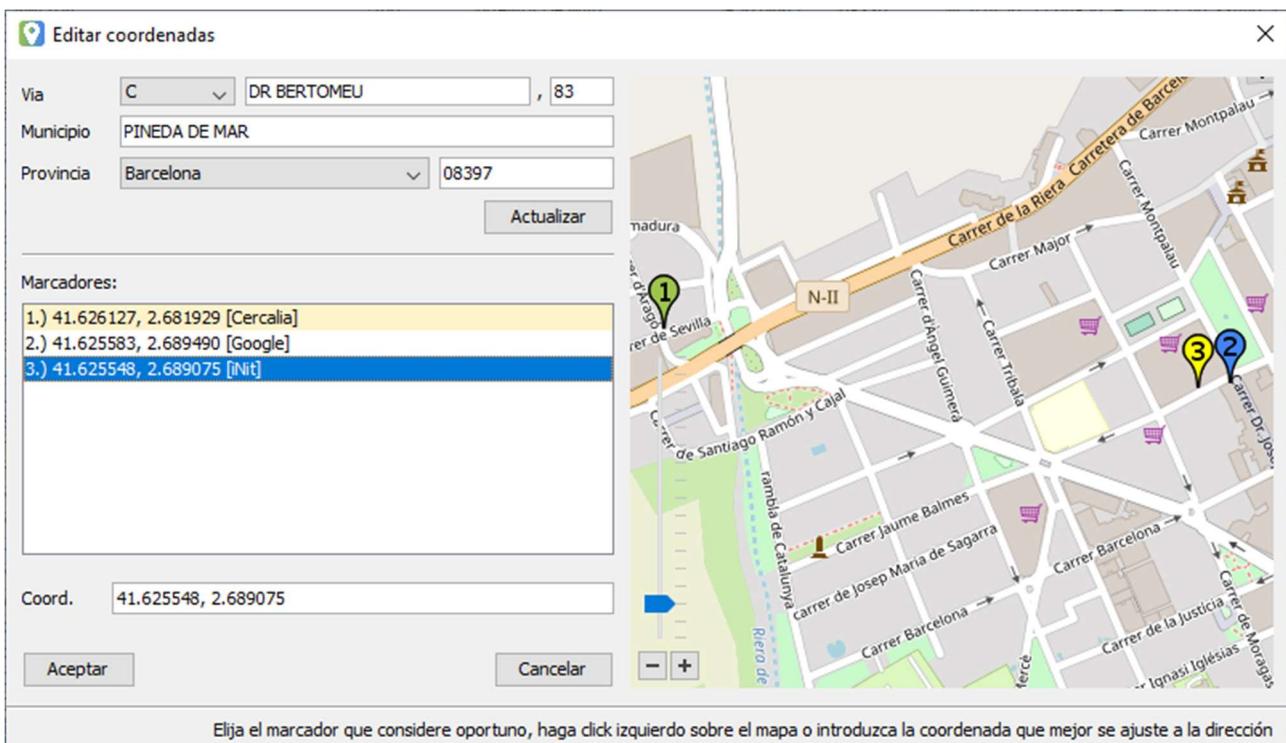


Ilustración 11. Ventana de edición de coordenadas (Manual)



Al pulsar en Aceptar, aparecerá una ventana de confirmación (*Ilustración 12*). A partir de este punto, los cambios serán irreversibles, almacenándose en la base de datos la nueva coordenada para la dirección que se estuviese editando.

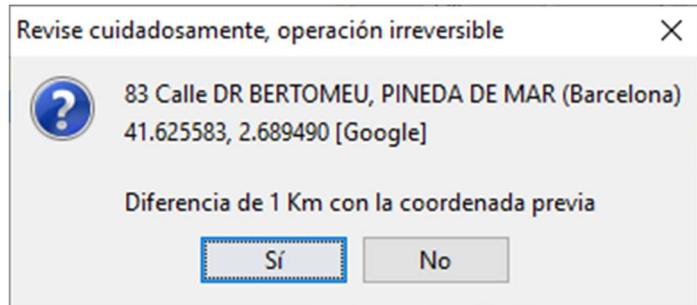


Ilustración 12. Confirmación de cambios

En la imagen se informa de que, tras seleccionar finalmente la coordenada proporcionada por Google, la distancia con la anterior es menor a 1 kilómetro.

3.3.- Distancias

La pantalla de distancias se comporta de una manera muy similar a la de direcciones. Cuenta con las mismas herramientas y una tabla ordenable.

Estado	Lat. origen	Lon. origen	Lat. destino	Lon. destino	Fecha inserción	Distancia (Km)	Duración (Min)	Proveedor	Porcentaje tiempo
Validado	N 28° 12' 45.52920"	W 14° 1' 14.73960"	N 28° 12' 46.73160"	W 14° 1' 12.73440"	06-09-2019	0.3	2	Manual	0.0
Validado	N 28° 12' 45.52920"	W 14° 1' 14.73960"	N 28° 12' 51.28560"	W 14° 1' 14.28600"	05-09-2019	0.5	3	Cercalia	0.0
Sin Validar	N 28° 12' 45.52920"	W 14° 1' 14.73960"	N 28° 29' 39.17400"	W 13° 51' 59.44320"	05-09-2019	44.9	40	Cercalia	0.075
Sin Validar	N 28° 12' 45.52920"	W 14° 1' 14.73960"	N 28° 30' 7.28640"	W 13° 51' 27.23040"	10-09-2019	46.5	45	Cercalia	0.044
Sin Validar	N 28° 12' 46.73160"	W 14° 1' 12.73440"	N 28° 12' 51.28560"	W 14° 1' 14.28600"	27-08-2019	0.5	3	Cercalia	0.5
Sin Validar	N 28° 12' 46.73160"	W 14° 1' 12.73440"	N 28° 29' 39.17400"	W 13° 51' 59.44320"	05-09-2019	44.8	40	Cercalia	0.05
Sin Validar	N 28° 12' 51.28560"	W 14° 1' 14.28600"	N 28° 12' 46.73160"	W 14° 1' 12.73440"	05-09-2019	0.2	2	Cercalia	0.5
Validado	N 28° 12' 51.28560"	W 14° 1' 14.28600"	N 28° 29' 39.17400"	W 13° 51' 59.44320"	05-09-2019	44.9	40	Cercalia	0.0
Sin Validar	N 28° 12' 55.24200"	W 14° 1' 5.81880"	N 28° 13' 2.51400"	W 14° 1' 3.51840"	05-09-2019	0.4	1	Cercalia	0.0
Sin Validar	N 28° 12' 55.24200"	W 14° 1' 5.81880"	N 28° 29' 39.17400"	W 13° 51' 59.44320"	05-09-2019	44.8	39	Cercalia	0.064
Sin Validar	N 28° 12' 55.24200"	W 14° 1' 5.81880"	N 28° 30' 10.11600"	W 13° 52' 14.80440"	05-09-2019	46.7	43	Cercalia	0.034
Sin Validar	N 28° 12' 55.24200"	W 14° 1' 5.81880"	N 28° 30' 21.74040"	W 13° 52' 42.25800"	05-09-2019	47.3	45	Cercalia	0.022
Sin Validar	N 28° 12' 55.24200"	W 14° 1' 5.81880"	N 28° 36' 37.43280"	W 13° 55' 43.68360"	05-09-2019	55.7	52	Cercalia	0.019
Sin Validar	N 28° 13' 2.51400"	W 14° 1' 3.51840"	N 28° 25' 24.50280"	W 14° 0' 52.39440"	10-09-2019	26.7	27	Cercalia	0.055
Sin Validar	N 28° 13' 2.51400"	W 14° 1' 3.51840"	N 28° 29' 33.17640"	W 13° 57' 18.85680"	05-09-2019	37.0	37	Cercalia	0.054
Sin Validar	N 28° 13' 2.51400"	W 14° 1' 3.51840"	N 28° 30' 7.22880"	W 13° 50' 54.69720"	10-09-2019	47.3	45	Cercalia	0.044

Ilustración 13. Tabla de distancias

Con respecto a sus columnas:

- **Estado:** Una ruta puede contar solo con 4 estados con sus colores correspondientes, Autovalidado (azul), Validado (verde), Sin validar (naranja) y Sin alternativas (blanco).
- **Lat. origen:** Latitud de la coordenada de origen.
- **Lon. origen:** Longitud de la coordenada de origen.



- **Lat. destino:** Latitud de la coordenada de destino.
- **Lon. destino:** Longitud de la coordenada de destino.
- **Fecha de inserción:** Fecha de la última modificación de cualquiera de los campos.
- **Distancia:** Distancia en kilómetros (mediante carreteras) asignada a las coordenadas.
- **Duración:** Distancia en minutos (vehículo por carretera) asignada a las coordenadas.
- **Proveedor:** Organización que ha proporcionado la distancia y duración
- **Porcentaje tiempo:** Solo cuenta con valor para las distancias sin validar, siendo la media porcentual de la diferencia entre la distancia/duración actual y sus alternativas (máx. 1).

El mayor cambio aparente con respecto a la pestaña de direcciones es su ventana de edición, que es una versión simplificada de la de coordenadas.

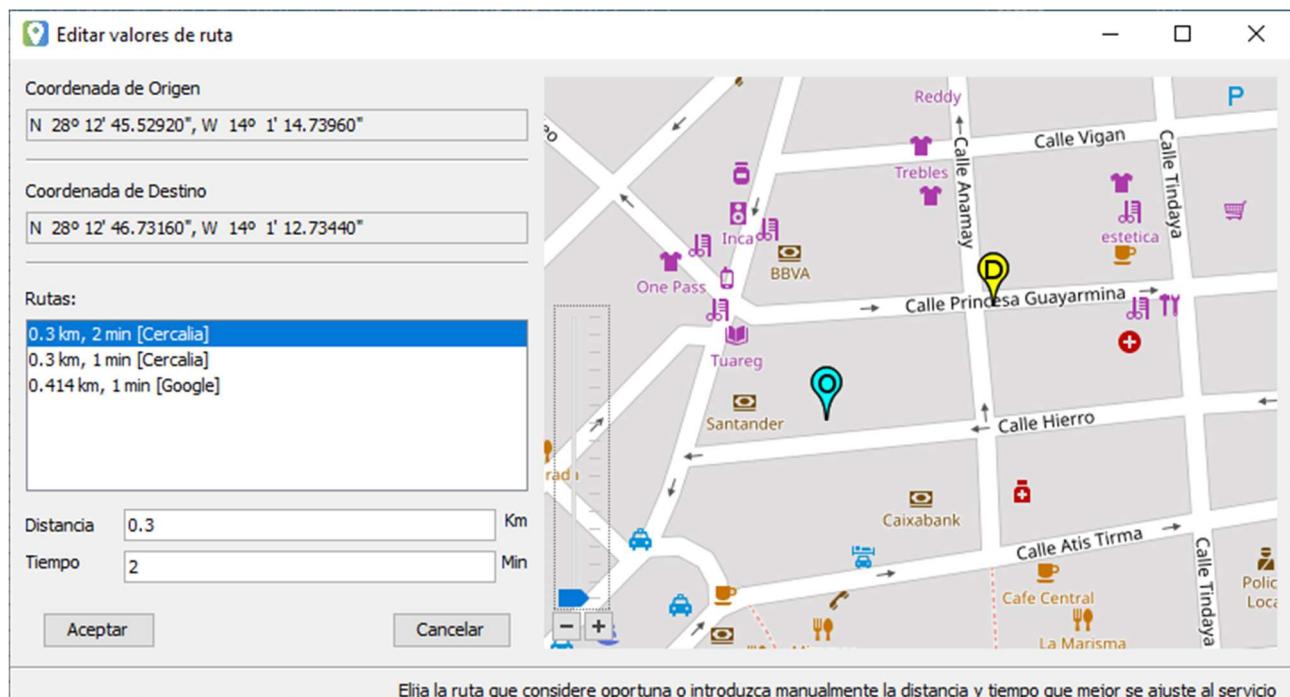


Ilustración 14. Ventana de edición de rutas

Nos muestra las coordenadas de origen (O cian) y destino (D amarilla), y en este caso el mapa no es interactivo, por lo que deberemos elegir una de las propuestas, o introducir la distancia en kilómetros (decimales) y minutos (enteros).

Tras aceptar, nos aparecerá también su correspondiente ventana de confirmación.

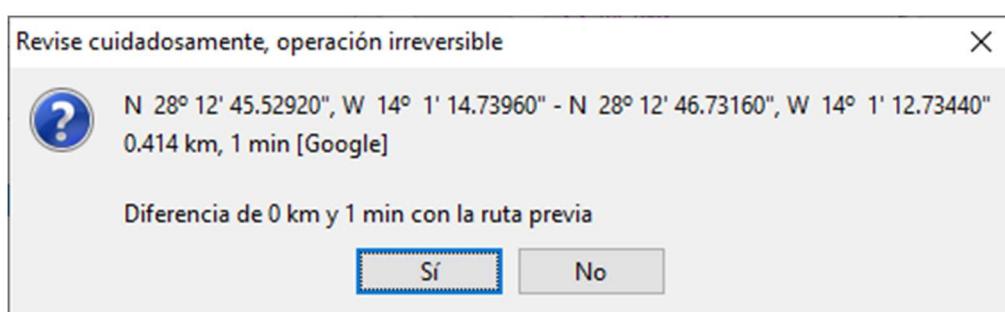


Ilustración 15. Ventana de confirmación de cambios



3.4.- Usuarios

Este apartado del programa sirve para asociar las zonas en que un usuario puede operar. Haciendo doble click sobre un usuario se puede cambiar su nombre, y si se selecciona, aparecerán sus zonas asociadas en la tabla central y las restantes en la tabla de la derecha.

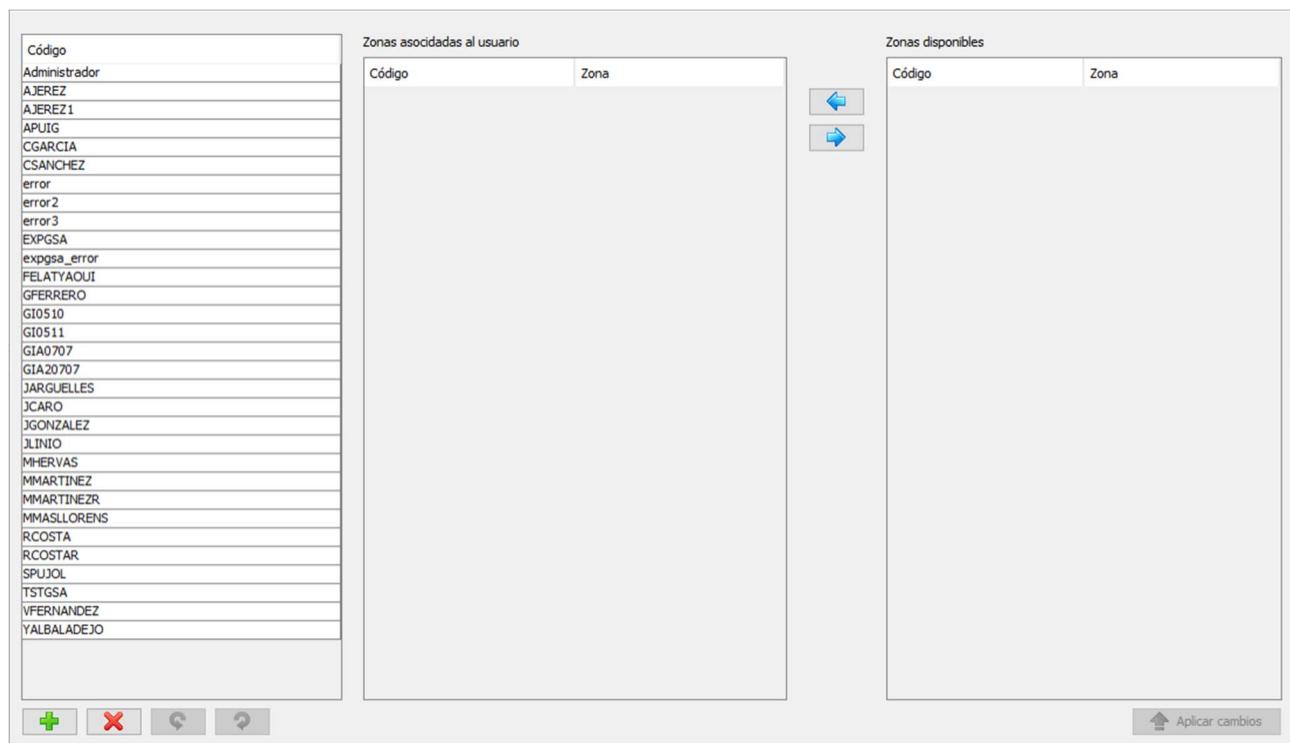


Ilustración 16. Pestaña de usuarios

Con los botones de abajo a la izquierda, se pueden crear nuevos usuarios, eliminarlos o deshacer cualquier acción. Sin embargo, los cambios no se producirán de forma definitiva hasta que se pulse el botón Aplicar cambios.

Haciendo click derecho sobre uno de los elementos de la tabla de la izquierda se pueden realizar las mismas operaciones, además de duplicar dicho usuario, lo que reduce el número de pasos a la hora de crear nuevos usuarios.

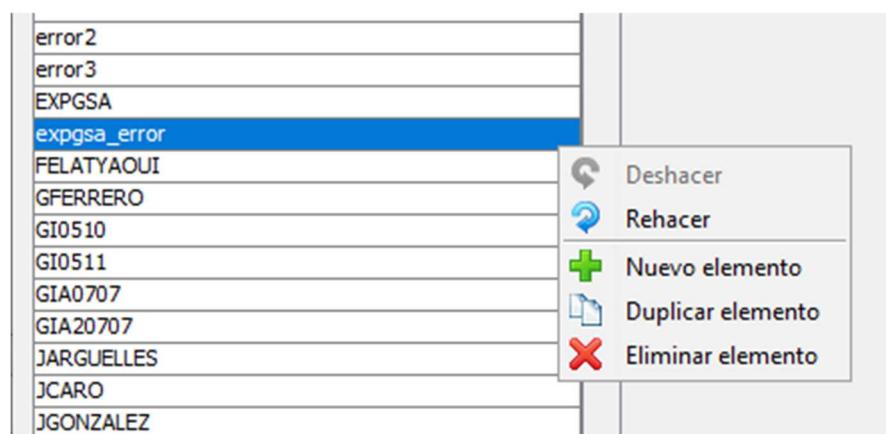


Ilustración 17. Menú desplegable



Código	Zona
Administrador	
AJEREZ	Igualada
AJEREZ1	Alt Penedes
APUIG	Alcaniz
CGARCIA	Barcelona
CSANCHEZ	Sallent
error	Martorell
error2	Molins de Rei Nord
error3	Barbastro
EXPGSA	Calatayud
expgsa_error	Alt Empordà
FELATYAOUI	Baix Empordà
GFERRERO	Girona
GI0510	Tarragona Privats
GI0511	Zaragoza
GIA0707	
GIA20707	
JARGUELLES	
JCARO	
JGONZALEZ	
JLNUJO	
MHERVAS	
MMARTINEZ	
MMARTINEZR	
MMASLORENS	
RCOSTA	
RCOSTAR	
SPUJOL	
TSTGSA	
VFERNANDEZ	
YALBALADEJO	
NUEVO_USUARIO	

Código	Zona
AN	Igualada
AP	Alt Penedes
AZ	Alcaniz
BC	Barcelona
BG	Sallent
BL	Martorell
BLM	Molins de Rei Nord
BR	Barbastro
CT	Calatayud
EA	Alt Empordà
EB	Baix Empordà
GI	Girona
TP	Tarragona Privats
ZG	Zaragoza

Código	Zona
AR	Pont de Suert
AU	La Seu
BE	Berga
CM	Gran Canaria
CP	Gran Canaria
FP	Fuerteventura
FR	Fuerteventura
GP	La Gomera
GX	Garrotxa
HI	Hierro
HU	Huesca
JA	Jaca
LC	Puigcerda
LG	La Gomera
LHM	Molins de Rei
LZ	Lanzarote
MN	Maresme
MT	Mataró
OS	Gurb
P1	Privats Metropolita
P2	Privats Rodalies
P3	Privats Girona
P4	Mataró Privats
PJ	Trem
PM	Palma de Mallorca
PS	Sort
PT	Privats Tarra- Reus
RI	Ripollés
RZ	Rural Zaragoza
SI	Selva interior
SM	Selva Marítima
SO	Solsona
TC	Santa Cruz

Ilustración 18. Editando usuarios

En este ejemplo se puede observar que se ha editado al usuario error, o bien su nombre o alguna de sus zonas (**naranja**). JGONZALEZ está seleccionado y se muestran sus zonas, al que se le han añadido las zonas BG, BLM y BR en **verde** y eliminado las que están en **naranja** en la tabla adyacente. Además de lo anterior, al aplicar cambios se eliminarán RCOSTA y RCOSTAR (**rojo**) y se añadirá a NUEVO_USUARIO (**verde**).

Sobra decir que al abandonar esta pestaña o cerrar la aplicación, todos los cambios indicados por colores no se producirán, ya que se debe pulsar previamente el botón de Aplicar cambios.

3.5.- Zonas y bases

Consta de dos tablas independientes con un funcionamiento más simplificado que el de usuarios. El único aspecto destacable es la necesidad de asignar una base a una zona mediante su desplegable.

17079	C	Curiol	22	Quart	17	724	GI
17114	C	bestraca	0	Olot	17	724	GX
17147	C	Cerdanya	0	Campdevánol	17	724	CT
17193	C	can romaní	0	Santa Coloma ...	17	724	EA
22048	C	Virgen del Pilar	3	BARBASTRO	22	724	EB
22125	C	Badalona	9	HUESCA	22	724	FP
22130	PLG	Industrial	0	JACA	22	724	FR
35017	C	Manuel Velázq...	2	Fuerteventura	35	725	GI
43206	C	Plom	7	Tarragona	43	724	GP
44013	PLG	LAS HORCAS N...	0	ALCAÑIZ	44	724	GX
50297	C	CHARLUCA	0	CALATAYUD	46	724	CT
5067	PLG	EL PORTAZGO ...	0	ZARAGOZA	46	724	ZG

Ilustración 19. Tabla de bases



4.- Alertas del sistema

Durante la ejecución o inicio de la aplicación es probable que aparezca alguna alerta. Para el caso del sistema de validación automática, estas alertas pueden desencadenar la pausa del mismo, o su total parada.

Existe otra instancia de esta aplicación

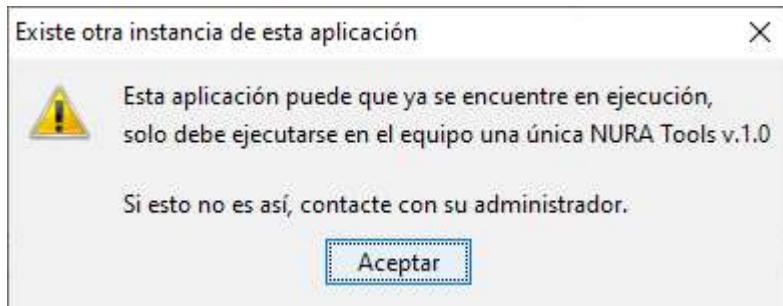


Ilustración 20. Existe otra instancia de esta aplicación

Es una alerta común, ocurre al tratar de abrir dos o más aplicaciones de NURA Tools.

Excepción SQL relacionada con la base de datos

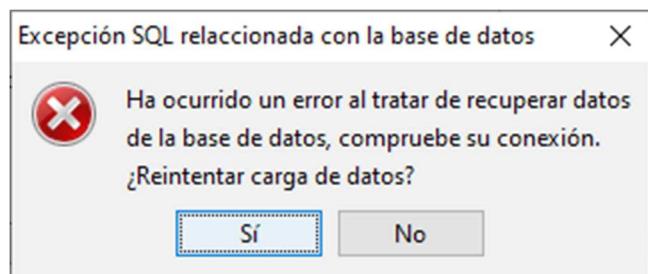


Ilustración 21. Excepción SQL (inicio)

Este error solo puede ocurrir al iniciar el sistema, e informa de problemas de conexión al sincronizar con la base de datos. Permite reintentar o cerrar la aplicación.

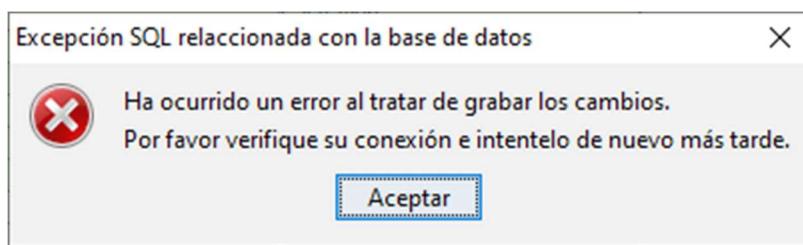


Ilustración 22. Excepción SQL

Este otro ocurre durante la ejecución del programa, impide el funcionamiento del proceso de fondo pausándolo o impide grabar nuevos datos. Para este caso no hay opción de reintentar, simplemente se pulsa en Aceptar y se prueba de nuevo en la ventana que hubiese generado el error o se reanuda manualmente la autovalidación.



Error de conexión remota

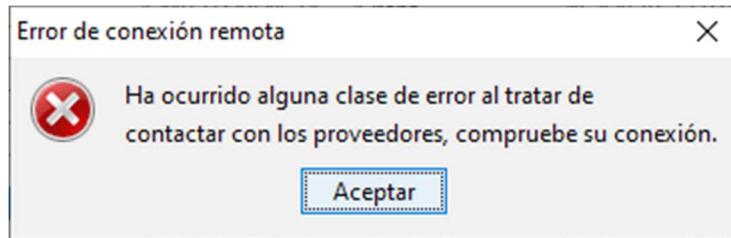


Ilustración 23. Error de conexión remota

Este error ocurre también por problemas de conexión con los servidores de los proveedores. El problema de conexión no tiene por qué ser necesariamente nuestro, aunque sea lo más probable.

Error grave de interpretación

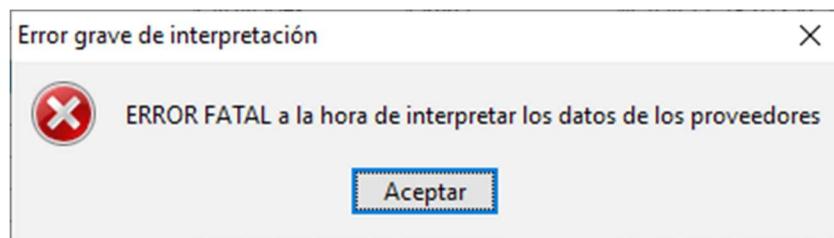


Ilustración 24. Error grave de interpretación

Este error no tiene solución por parte del usuario y es extremadamente extraño que pueda llegar a producirse. Ocurriría en el caso de que los proveedores alterasen de alguna manera el formato de los mensajes mediante los cuales la aplicación se comunica con ellos.

La cuenta de Google no tiene saldo suficiente

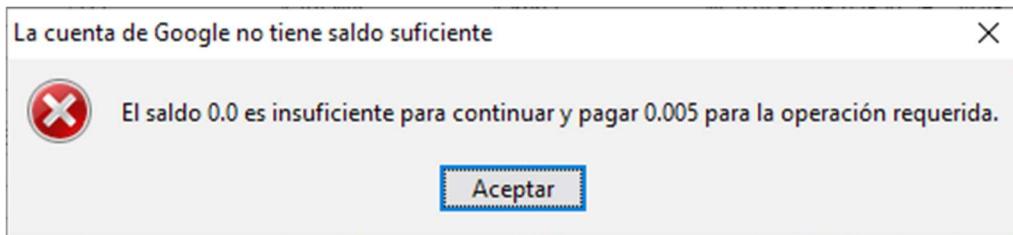


Ilustración 25. Google no tiene saldo suficiente

No es un error como tal, pero impide que continuemos con la validación. Google es un servicio de pago que funciona mediante prepago, si nuestra cartera está vacía saltará este error. No obstante, con la creación de una nueva cuenta, Google regala 300\$ de saldo que caducan pasados 12 meses, y que permite llevar a cabo unas 60.000 consultas.



5.- FAQ

Al abrir la aplicación la tabla de direcciones está vacía ¿por qué?

Lo ideal es que la tabla esté vacía, pues indica que no hay direcciones que necesiten de validación manual. Al pulsar en el botón de autovalidación, tras un breve periodo de tiempo, es probable que comiencen a aparecer. También puede ser que todas estén validadas y no exista ninguna dirección sin alternativas.

La autovalidación está en marcha, pero no aparecen distancias a validar ¿por qué?

La aplicación está preparada para que compruebe primero las direcciones. Hasta que no quede ninguna dirección, no comenzará a revisar las distancias. No obstante, siempre puedes mostrar las distancias sin alternativas y hacer click en una de ellas para generarla con prioridad.

¿Qué es un proveedor?

Los proveedores, en concreto para esta aplicación son Cercalia y Google. El programa consulta a sus sistemas acerca de las coordenadas asociadas a una dirección, así como la distancia en tiempo y espacio entre dos coordenadas geográficas.

¿Por qué algunas de las coordenadas no tienen minutos y segundos?

Las coordenadas que aparecen dentro de las ventanas de editar son coordenadas decimales (DD). Si se desea introducir una coordenada con grados, minutos y segundos (DMS) puedes utilizar un conversor online, aunque es preferible hacer click derecho en el mapa para capturar una coordenada manualmente.

¿Por qué no se ven las rutas de las tablas de distancias?

Las rutas gráficas es información adicional que hay que pedir a los proveedores. Esto conlleva un correspondiente retardo que ralentizaría la aplicación. No obstante, al hacer doble click sobre una ruta de Google, como la de la *Ilustración 14*, se desplegará la misma en el navegador predeterminado.

¿Por qué no puedo asociar zonas a un usuario?

Se debe tener seleccionado a un usuario que no esté marcado para eliminar.

¿Por qué no puedo editar una zona/base/usuario?

Cuando un elemento se marca para eliminar, éste se vuelve inalterable, no tiene sentido editar algo que se va a borrar. Si no deseaba borrarlo y deshacer no es una opción, pulsar de nuevo en eliminar lo desmarcará.

¿Por qué no puedo asignar una zona recién creada a una base?

Si dicha zona se ha creado, pero no se ha pulsado en el botón de aplicar cambios correspondientes, no se puede hacer uso de tal, por parte de una base o asignar a un usuario, hasta que se apliquen cambios en la tabla de zonas y se grabe el correspondiente registro en la base de datos.

