

**UNIVERSIDAD EUROPEA DE MADRID**  
**ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO**

**ÁREA INGENIERÍA INDUSTRIAL**

**MÁSTER UNIVERSITARIO EN**  
**INGENIERÍA INDUSTRIAL**

**TRABAJO FIN DE MASTER**  
**DESARROLLO DE UN SOFTWARE PARA**  
**OPTIMIZACIÓN DE RUTAS Y DEL PLAN DE**  
**NEGOCIO PARA SU LANZAMIENTO**

**Alumno: D. Jorge Pérez Trueba**  
**Director: D. Miguel Florez de la Colina**

**ENERO 2025**

Desarrollo de un software para optimización de rutas y del plan de negocio para su lanzamiento.  
Jorge Pérez Trueba

---

**TÍTULO:** Desarrollo de un software para optimización de rutas y del plan de negocio para su lanzamiento.

**AUTOR:** Jorge Pérez Trueba

**DIRECTOR DEL PROYECTO:** Miguel Florez de la Colina

**FECHA:** 23 de enero de 2025

## RESUMEN

El presente trabajo desarrolla un software para la optimización de rutas en coche y transporte público, que permite generar paquetes de trabajo diarios a partir de una hoja de Excel con relaciones de direcciones. Posteriormente, se elabora un plan de negocio para la creación de una empresa destinada a comercializar este software, con el objetivo de obtener los recursos financieros necesarios para su lanzamiento.

Los objetivos principales del proyecto son, por un lado, diseñar un software sencillo, intuitivo y enfocado en la sostenibilidad, capaz de cubrir necesidades desatendidas en el mercado; y por otro, evaluar la viabilidad de su comercialización y el retorno de la inversión requerida.

El software se ha desarrollado en Python, utilizando la API Directions de Google Maps y librerías como Pandas y Datetime. En el plan de negocio, se ha realizado un análisis del mercado, identificación de segmentos objetivo, definición de precios, selección de canales de venta, previsión de ventas y estimación de costes.

El análisis financiero muestra que la empresa representa una oportunidad única de inversión al combinar rentabilidad, crecimiento y sostenibilidad. Con una inversión inicial de 5.000€, se proyecta un VAN de 4.579,26€ con una tasa de retorno del 10% y una TIR del 39%, excluyendo el valor futuro estimado de la empresa en 2028.

Además de reducir emisiones, ruido y problemas de tráfico en áreas urbanas, el software ofrece un punto de partida sólido para emprender su comercialización en plataformas como GitHub o Gumroad, destacando por su viabilidad comprobada.

**Palabras clave:** software, Python, rutas, negocio, comercialización, sostenibilidad.

## ABSTRACT

This master's final project develops a software for the optimisation of car and public transport routes, which allows the generation of daily work packages from an Excel spreadsheet that has addresses on its files. Subsequently, a business plan is developed for the creation of a company to commercialise this software, with the aim of raising the necessary financial resources to launch it.

The main aims of the project are, on the one hand, to design a simple, intuitive and sustainability-focused software capable of covering unattended needs in the market; and on the other hand, to assess the feasibility of its commercialisation and the return of the required investment.

The software has been developed in Python, using the Directions API of Google Maps and libraries such as Pandas and Datetime. In the business plan, a market analysis, identification of target segments, definition of prices, selection of sales channels, sales forecasts and cost estimations have been carried out.

The financial analysis shows that the company represents a unique investment opportunity by combining profitability, growth and sustainability. With an initial investment of €5,000, an NPV of €4,579.26 is projected with a rate of return of 10% and an IRR of 39%, excluding the estimated future value of the company in 2028.

In addition to reducing emissions, noise and traffic problems in urban areas, the software offers a solid starting point for commercialisation on platforms such as GitHub or Gumroad, and stands out for its proven viability.

**Key words:** software, Python, routes, business, commercialization, sustainability.

# Índice

<b>RESUMEN</b> .....	3
<b>ABSTRACT</b> .....	4
<b>Capítulo 1. Introducción.</b> .....	10
1.1 Relevancia de la eficiencia en la planificación de trayectos. ....	10
1.2 Planteamiento del problema. ....	10
1.3 Objetivos del proyecto. ....	11
1.4 Estructura del proyecto. ....	12
<b>Capítulo 2. Estado del arte de los softwares para optimización de rutas.</b> .....	13
<b>Capítulo 3. Desarrollo y explicación del código de optimización de rutas generado.</b> 15	
3.1 Información de entrada.....	15
3.2 Justificación del entorno de programación seleccionado. ....	16
3.3 Bibliotecas de Python utilizadas. ....	17
3.3.1 Biblioteca estándar de Python.....	17
3.3.2 Pandas .....	19
3.3.3 Googlemaps.....	20
3.3.4 Datetime.....	20
3.4 Interfaces de programación de aplicaciones (APIs) empleadas. ....	21
3.4.1 API Directions de GoogleMaps .....	22
3.4.2 Otra alternativa, APIs opencagedata y openrouteservice.....	24
3.5 Estructura del código.....	26
3.5.1 Importación de bibliotecas y otras configuraciones iniciales:.....	26
3.5.2 Almacenamiento de la información de entrada. Cálculo de distancias y tiempos en transporte público y generación de un archivo .xlsx con esta información. ....	27
3.5.3 Determinación de los emplazamientos a los que se puede asistir en transporte público y de aquellos en los que se utilizará el coche. ....	28
3.5.4 Cálculo de las asignaciones de emplazamientos que se cubrirán en transporte público y en coche.....	29
3.5.5 Obtención de archivos Excel con el reparto generado y con las visitas por realizar. 31	
3.6 Información de salida facilitada tras la ejecución. ....	31

---

<b>Capítulo 4. Plan de negocio.....</b>	<b>33</b>
4.1 Oportunidad.....	33
4.1.1 Formulación de la oportunidad. ....	33
4.1.2 Descripción del mercado. ....	34
4.1.3 Modelo de negocio objetivo.....	36
4.2 Segmentación de clientes.....	37
4.3 Propuesta de valor. ....	38
4.3.1 Atributos diferenciales de la propuesta de valor.....	38
4.3.2 Descripción de productos y servicios. ....	41
4.3.3 Plan de desarrollo y lanzamiento de productos y servicios. ....	43
4.3.4 Modelos de precios e ingresos. ....	46
4.4 Modelo comercial y marketing. ....	50
4.4.1 Plan de comunicación. ....	50
4.4.2 Canales de venta.....	51
4.5 Modelo de relación con clientes.....	53
4.6 Modelo de operaciones.....	54
4.7 Partners y alianzas.....	56
4.8 Gestión de talento.....	58
4.9 Sistemas de información. ....	62
4.10 Modelo económico. ....	65
4.10.1 Flujos de caja. ....	65
4.10.2 Balance de situación. ....	66
4.10.3 Cuenta de resultados. ....	70
4.11 Resumen ejecutivo.....	71
<b>Capítulo 5. Conclusión y futuras líneas de trabajo.....</b>	<b>74</b>
5.1 Conclusiones. ....	74
5.2 Futuras líneas de trabajo. ....	76
<b>ANEXOS.....</b>	<b>78</b>
Anexo I. Código de Python desarrollado. ....	78
Anexo II. Amortización del préstamo.....	89
Anexo III. Tablas con los flujos de caja de los tres primeros años. ....	91
Anexo IV. Asientos contables de los tres primeros años. ....	93
<b>BIBLIOGRAFÍA.....</b>	<b>97</b>

# Índice de Figuras

Fig. 1.- Grafo de partida que permite determinar mediante el algoritmo de Dijkstra el camino más corto desde Seattle a distintas ciudades de Estados Unidos. Fuente: Albalate, D. (2021, agosto 3). Algoritmo Dijkstra con giros permitidos. Medium. <a href="https://medium.com/">https://medium.com/</a> .....	13
Fig. 2.- Formato del archivo de Excel "reparto.xlsx" con la información de entrada a facilitar por el usuario. Fuente de elaboración propia.....	16
Fig. 3.- Plan de pricing de la API Directions de Google Maps. Fuente Google. "Uso y facturación de la API de Directions" Google Maps Platform Documentation.....	22
Fig. 4.- Planes que ofrece la API Openrouteservice, todos ellos gratuitos. Fuente: OpenRouteService "Plans and Pricing" OpenRouteService. ....	24
Fig. 5.- Planes que ofrece Opencage, entre cuyas APIs se encuentra Opencagedate. Fuente: OpenCage. "Pricing" OpenCage. ....	25
Fig. 6.- Archivo reparto_con_datos_transporte_público.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx. Fuente: elaboración propia, generado a partir de la ejecución del código. ....	27
Fig. 7.- Hoja Andando del archivo distancias.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx. Fuente: elaboración propia, generado a partir de la ejecución del código. ....	28
Fig. 8.- Hoja Coche del archivo distancias.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx. Fuente: elaboración propia, generado a partir de la ejecución del código. ....	29
Fig. 9.- Hoja "Andando" del archivo Reparto_generado.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx tras especificar el número de paquetes y emplazamientos por paquete. Fuente: elaboración propia, generado a partir de la ejecución del código. ....	32
Fig. 10.- Hoja "Coche" del archivo Reparto_generado.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx tras especificar el número de paquetes y emplazamientos por paquete. Fuente: elaboración propia, generado a partir de la ejecución del código. ....	32
Fig. 11.- Archivo Reparto_generado.xlsx, en el que, tras la ejecución, determinados los paquetes de trabajo, se recogen los emplazamientos que han quedado sin asignar. Fuente: elaboración propia, generado a partir de la ejecución del código.....	32
Fig. 12.- Análisis DAFO de la empresa que lanzará el software desarrollado. Fuente: Canva. "Diseño DAGa-GVaCyl" <a href="https://www.canva.com">https://www.canva.com</a> (plantilla).....	35
Fig. 13.- Análisis DAFO de la empresa que lanzará el software desarrollado. Fuente: Canva. "Diseño DAGa-OITzgY" <a href="https://www.canva.com">https://www.canva.com</a> (plantilla).....	36
Fig. 14.- Business model canvas de la empresa EcoRutas, que lanzará al mercado el software desarrollado. Fuente: elaboración propia. ....	37
Fig. 15.- Cuadrante mágico de Gartner de los softwares de optimización de rutas. Fuente: LanRoute. "Cuadrante Mágico de Gartner". <a href="https://lanroute.com/">https://lanroute.com/</a> .....	39
Fig. 16.- Principal competidor del producto desarrollado. Fuente: Route4Me. (n.d.). Planificación y optimización de rutas. <a href="https://route4me.com">https://route4me.com</a> .....	40
Fig. 17.- Curva de valor que compara las características del producto ofertado con las de su principal competidor, Route4Me. Fuente: elaboración propia.....	41
Fig. 18.- Resumen de la metodología scrum. Fuente: Lacort Peña, B. (2024-2025). Apuntes de la asignatura Dirección de grandes proyectos. Universidad Europea de Madrid.....	44
Fig. 19.- Cronograma de desarrollo del MPV, lanzamiento e implementación de nuevas versiones. Fuente: elaboración propia.....	45
Fig. 20.- Determinación gráfica del punto de equilibrio a partir de los valores acumulados de margen bruto y costes fijos. Fuente: elaboración propia.....	49
Fig. 21.- Logo de la marca (EcoRutas). Fuente: elaboración propia.....	50

---

*Fig. 22.- Logos de Gumroad (izda.) y GitHub (dcha.), en cuyos marketplaces se venderá el software desarrollado. Fuentes: Gumroad. (n.d.). Sell what you know and see what sticks. <https://gumroad.com/> y GitHub. (n.d.). Where the world builds software. <https://github.com/>.....51*

*Fig. 23.- Descripción de las interacciones de los clientes con la empresa sintetizada en el "customer journey map". Fuente: elaboración propia .....53*

*Fig. 24.- CMI de EcoRutas que sintetiza los indicadores clave relacionados con los objetivos estratégicos de la empresa, las metas propuestas, iniciativas clave y periodicidades de evaluación. Fuente: elaboración propia .....55*

*Fig. 25.- Clientes promotores, pasivos y detractores en función de la puntuación del producto en la encuesta de satisfacción. Fuente: SignificaCloud. (n.d.). Voz del empleado, voz del cliente y NPS. <https://www.meaningcloud.com/es/blog/voz-del-empleado-voz-del-cliente-y-nps>..... 56*

*Fig. 26.- Planes de HERE Technologies. Fuente: HERE Technologies. (n.d.). Pricing plans for location services. <https://www.here.com/get-started/pricing> ..... 57*

*Fig. 27.- Organigrama propuesto para la empresa. Fuente: elaboración propia..... 58*

*Fig. 28.- : Asientos contables del año 2026. Fuente: elaboración propia ..... 94*

*Fig. 29.- Asientos contables del año 2027. Fuente: elaboración propia .....95*

*Fig. 30.- Asientos contables del año 2028. Fuente: elaboración propia ..... 96*

## Índice de Tablas

<i>Tabla 1.- Evolución de ventas prevista durante los tres primeros años de la empresa, 2026, 2027 y 2028. Fuente: elaboración propia.</i>	49
<i>Tabla 2.- Costes previstos de los distintos perfiles de los trabajadores de la empresa. Fuente: Impulsa Empresa. (n.d.). Coste del trabajador para la empresa <a href="https://www.impulsa-empresa.es/coste-trabajador-empresa/">https://www.impulsa-empresa.es/coste-trabajador-empresa/</a>.</i>	61
<i>Tabla 3.- Estimación de los gastos operativos en IT y comunicaciones de la empresa EcoRutas. Fuente: elaboración propia.</i>	64
<i>Tabla 4.- Estimación de la inversión necesaria en equipos informáticos de los trabajadores. Fuente: elaboración propia.</i>	64
<i>Tabla 5.- coeficientes de amortización lineal establecidos por la Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades. Fuente: España. (2014). Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades. Boletín Oficial del Estado (BOE), núm. 288, de 28 de noviembre de 2014, pp. 96939-97058. <a href="https://www.boe.es/buscar/doc.php?id=BOE-A-2014-12328">https://www.boe.es/buscar/doc.php?id=BOE-A-2014-12328</a>.</i>	64
<i>Tabla 6.- Balance de EcoRutas a 31 de diciembre de 2026. Fuente: elaboración propia.</i>	67
<i>Tabla 7.- Balance de EcoRutas a 31 de diciembre de 2027. Fuente: elaboración propia.</i>	67
<i>Tabla 8.- Balance de EcoRutas a 31 de diciembre de 2028. Fuente: elaboración propia.</i>	68
<i>Tabla 9.- Cuenta de resultados de EcoRutas del año 2026. Fuente: elaboración propia.</i>	70
<i>Tabla 10.- Cuenta de resultados de EcoRutas del año 2027. Fuente: elaboración propia.</i>	70
<i>Tabla 11.- Cuenta de resultados de EcoRutas del año 2028. Fuente: elaboración propia.</i>	70
<i>Tabla 12.- Amortizaciones e intereses mensuales del préstamo pedido por EcoRutas el primer año. Fuente: elaboración propia.</i>	90
<i>Tabla 13.- Flujos de caja de EcoRutas esperados para el año 2026. Fuente: elaboración propia.</i>	91
<i>Tabla 14.- Flujos de caja de EcoRutas esperados para el año 2027. Fuente: elaboración propia.</i>	91
<i>Tabla 15.- Flujos de caja de EcoRutas esperados para el año 2028. Fuente: elaboración propia.</i>	92

# Capítulo 1. Introducción.

## 1.1 Relevancia de la eficiencia en la planificación de trayectos.

La eficiencia en la planificación de trayectos es crucial para las empresas que dependen de la logística y el transporte, ya que tiene un impacto directo en la reducción de costos operativos, la mejora del servicio al cliente y la sostenibilidad ambiental.

Optimizar los trayectos permite minimizar el consumo de combustible y los tiempos de desplazamiento, lo que se traduce en ahorros significativos en gastos de transporte y mantenimiento de vehículos. Además, una planificación eficiente asegura entregas puntuales, fortaleciendo la confianza y la satisfacción del cliente, factores esenciales para mantener la competitividad en mercados cada vez más exigentes.

Desde una perspectiva ambiental, reducir distancias innecesarias y optimizar rutas disminuye la emisión de gases de efecto invernadero, como el dióxido de carbono (CO<sub>2</sub>), principal responsable del cambio climático. Además, una menor cantidad de vehículos en la carretera o el uso de trayectos más cortos contribuye a mitigar la contaminación del aire local, beneficiando la salud de las comunidades en áreas urbanas o altamente transitadas. La eficiencia en la planificación también fomenta el uso racional de los recursos, como el combustible fósil, cuya extracción y consumo generan un impacto ambiental significativo. Por otro lado, al reducir la necesidad de recorridos redundantes o ineficientes, se minimiza el desgaste de los vehículos, lo que prolonga su vida útil y reduce la generación de residuos asociados a piezas y materiales de desecho.

En el contexto actual, donde las normativas ambientales y la presión social exigen a las empresas adoptar prácticas sostenibles, la planificación de trayectos representa una herramienta estratégica para alinear los objetivos corporativos con los compromisos medioambientales. Este enfoque no solo refuerza la reputación de la empresa, sino que también contribuye activamente a la construcción de un futuro más sostenible y resiliente.

En conjunto, la eficiencia en la planificación de trayectos no solo mejora los resultados financieros de la empresa, sino que también fortalece su reputación y compromiso con prácticas responsables.

## 1.2 Planteamiento del problema.

En el contexto español, las pequeñas y medianas empresas constituyen el pilar fundamental de la economía, desarrollando muchas de ellas actividades como el reparto, el mantenimiento y la resolución de incidencias, donde los márgenes de tiempo son críticos. Todas estas empresas se enfrentan a un desafío crítico común, la optimización de las rutas diarias para realizar tareas en distintos emplazamientos.

En un entorno donde la eficiencia operativa es clave para mantener la competitividad, estas empresas a menudo carecen de herramientas accesibles y económicas que les permitan gestionar de manera eficiente sus recursos. Muchas veces, aunque reconocen la necesidad

de modernizar sus procesos, a menudo se sienten intimidados por soluciones tecnológicas complejas o costosas. Además, generalmente no tienen grandes beneficios, por lo que,

aunque interesados en mejorar su operativa, no están dispuestas a comprometer su presupuesto con grandes inversiones.

Por otro lado, el cumplimiento de los objetivos de sostenibilidad está moldeando el futuro de las empresas en España. Los entornos urbanos sufren graves problemas de congestión vehicular al que contribuyen las actividades a las que se dedican las mencionadas empresas, y también de contaminación del aire, especialmente por NOx. Los niveles de NOx llegan a superar en ocasiones los límites establecidos por la Unión Europea debido al tráfico rodado, lo que ha llevado en las grandes ciudades a implementar restricciones progresivas al tránsito de vehículos de combustión interna y a desarrollar protocolos de contaminación, que se activan en episodios de alta polución, restringiendo aún más el acceso de vehículos contaminantes a determinadas zonas centrales.

### 1.3 Objetivos del proyecto.

Este Trabajo Fin de Máster tiene varios objetivos, algunos relacionados con el desarrollo de una herramienta para la optimización de rutas que dé una solución a problemas a los que se enfrentan muchas empresas, que ya han sido comentados en el apartado anterior y otros asociados al negocio que se pretende llevar a cabo, consiste en el lanzamiento del producto a través de varias plataformas que posibilitan la compra a las empresas.

Desde la **perspectiva del producto desarrollado** los objetivos son:

- Desarrollar un software, a partir de un lenguaje de programación apropiado, para la optimización de las rutas diarias que efectúan las empresas con el objetivo de realizar tareas en distintos emplazamientos.
- Ofrecer este software como un producto sencillo, intuitivo, que no requiere conocimientos avanzados de informática y además a un precio accesible, capaz de satisfacer necesidades no cubiertas por otros softwares de optimización de rutas disponibles en el mercado.
- Ayudar a las empresas a mejorar sus procesos de planificación de rutas de sus empleados, ahorrándoles tiempo y reduciendo sus costes asociados al transporte entre los distintos emplazamientos.
- Fomentar prácticas de transporte más responsables mediante la inclusión de rutas en transporte público, contribuyendo a la reducción de emisiones de gases de efecto invernadero y otros gases contaminantes como el NOx, así como a mejorar el tráfico en entornos urbanos.

Desde el **punto de vista de la idea de negocio** se persiguen los siguientes objetivos:

- Analizar los segmentos de mercado que pueden estar interesados en el producto, determinar la mejor forma de darlo a conocer y buscar canales de venta apropiados.
- Fijar un precio adecuado comparando las características y nivel de servicio de los productos similares de los principales competidores.

- Estudiar si es posible generar un retorno a la inversión realizada por parte de los inversores a partir del plan de negocio y de una estimación de los flujos de caja.
- Realizar un resumen ejecutivo atractivo para posibles inversores que sea capaz de poner en valor el producto y permita comenzar el negocio a partir de su inversión.

#### 1.4 Estructura del proyecto.

El Trabajo Fin de Máster se estructura en cinco capítulos.

El **Capítulo 1. Introducción**, justifica la importancia que tiene el trabajo para satisfacer necesidades todavía no cubiertas satisfactoriamente de empresas entre cuyas actividades se encuentra la optimización de rutas.

El **Capítulo 2. Estado del arte de los softwares de optimización de rutas**, explica cómo se han abordado los problemas de optimización de rutas a lo largo del tiempo y las nuevas tendencias, marcadas por los avances de la tecnología, la informática e internet. Seguidamente se comparan los softwares más relevantes disponibles en el mercado.

El **Capítulo 3. Desarrollo y explicación del código de optimización de rutas generado**, describe las bibliotecas, funciones y APIs (Interfaz de Programación de Aplicaciones) a las que se recurre para poder crear una herramienta que permite optimizar las rutas a seguir los trabajadores en su día a día, cómo se estructura el código, la información de entrada que es necesario facilitar a la herramienta y la información de salida, que incluye los trayectos diarios a realizar en coche o transporte público por parte de los trabajadores.

El **Capítulo 4. Plan de negocio**, analiza el valor que puede genera la herramienta en el mercado, determina los segmentos objetivo y los canales de venta, y desarrolla un modelo económico a partir de los flujos de caja previsibles durante los primeros años. Culmina con un resumen ejecutivo, que trata de atraer inversión que facilite la implementación del negocio y mejorar el producto progresivamente.

El **Capítulo 5. Conclusión y futuras líneas de trabajo**, además de sintetizar los aspectos más relevantes del trabajo incluye posibles futuras líneas de trabajo, entre las que destacan la mejora del producto mediante sucesivas versiones que incluyen tiempos acumulados invertidos por los trabajadores en los desplazamientos, la bicicleta como alternativa y la estimación del ahorro de emisiones de CO<sub>2</sub>; y un plan para lograr posicionarse en el mercado nacional.

## Capítulo 2. Estado del arte de los softwares para optimización de rutas.

Los softwares de optimización de rutas han evolucionado significativamente desde sus inicios, impulsados por avances tecnológicos y la creciente necesidad de eficiencia en la logística. En sus primeras etapas, estas herramientas se basaban en algoritmos básicos como el de Dijkstra y otros métodos heurísticos diseñados para encontrar rutas más cortas en redes de transporte relativamente simples. Estas soluciones estaban limitadas por la capacidad de los sistemas computacionales de la época y eran más aplicables a problemas de pequeña escala.

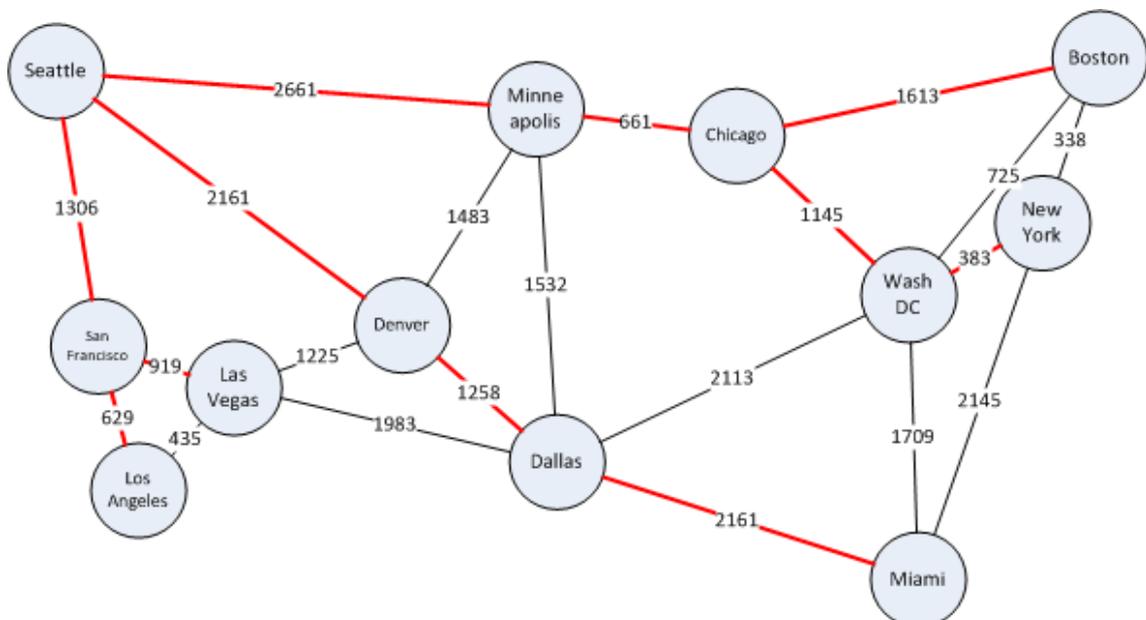


Fig. 1.- Grafo de partida que permite determinar mediante el algoritmo de Dijkstra el camino más corto desde Seattle a distintas ciudades de Estados Unidos. Fuente: Albalade, D. (2021, agosto 3). Algoritmo Dijkstra con giros permitidos. Medium. <https://medium.com/>

Con el avance de la computación y la llegada de los sistemas de información geográfica (GIS), los softwares de optimización integraron datos espaciales, permitiendo análisis más detallados y realistas. Esto significó un salto cualitativo, ya que las herramientas comenzaron a tener en cuenta variables como condiciones de tráfico, restricciones de vehículos y horarios específicos, mejorando considerablemente la precisión de las soluciones. Además, la proliferación de dispositivos móviles y GPS durante la década de 2000 facilitó la implementación de rutas optimizadas en tiempo real, llevando estas tecnologías directamente a las manos de los conductores.

En la actualidad, los softwares de optimización de rutas han adoptado tecnologías emergentes para abordar los desafíos del mundo moderno. Una de las tendencias más notables es la integración de inteligencia artificial (IA) y aprendizaje automático (machine learning). Estas tecnologías permiten a los sistemas aprender de los datos

---

históricos y en tiempo real para predecir patrones de tráfico, identificar cuellos de botella y adaptarse dinámicamente a cambios inesperados, como accidentes o condiciones meteorológicas adversas.

Otra tendencia clave es el uso de big data para enriquecer los análisis. Los softwares ahora incorporan datos provenientes de múltiples fuentes, como sensores IoT, redes sociales y aplicaciones de tráfico, para ofrecer soluciones más robustas y personalizadas. Este enfoque permite gestionar operaciones logísticas complejas, como entregas de última milla, que requieren una precisión y flexibilidad extremas.

Además, la sostenibilidad se ha convertido en un foco importante. Las nuevas herramientas de optimización están diseñadas para reducir el consumo de combustible y las emisiones de carbono, alineándose con los objetivos globales de sostenibilidad. Esto se logra mediante algoritmos que priorizan rutas ecológicas o maximizan la utilización de vehículos eléctricos.

Por último, la creciente demanda de entregas rápidas y personalizadas ha impulsado el desarrollo de plataformas que integran la optimización de rutas con la experiencia del cliente. Esto incluye aplicaciones que notifican en tiempo real a los usuarios sobre el progreso de sus pedidos y herramientas que permiten a los conductores comunicarse directamente con los clientes para resolver problemas logísticos.

## Capítulo 3. Desarrollo y explicación del código de optimización de rutas generado.

Un código de programación es un conjunto de instrucciones escritas en un lenguaje de programación que un ordenador puede interpretar y ejecutar. Estas instrucciones le indican a la computadora qué tareas debe realizar, desde operaciones simples como sumar dos números hasta procesos complejos como manejar bases de datos, generar gráficos o entrenar modelos de inteligencia artificial.

El código se escribe en lenguajes de programación, como Python, C++, o Java, que actúan como una interfaz entre el ser humano y el ordenador, ya que permiten expresar ideas de manera estructurada y comprensible para ambas partes.

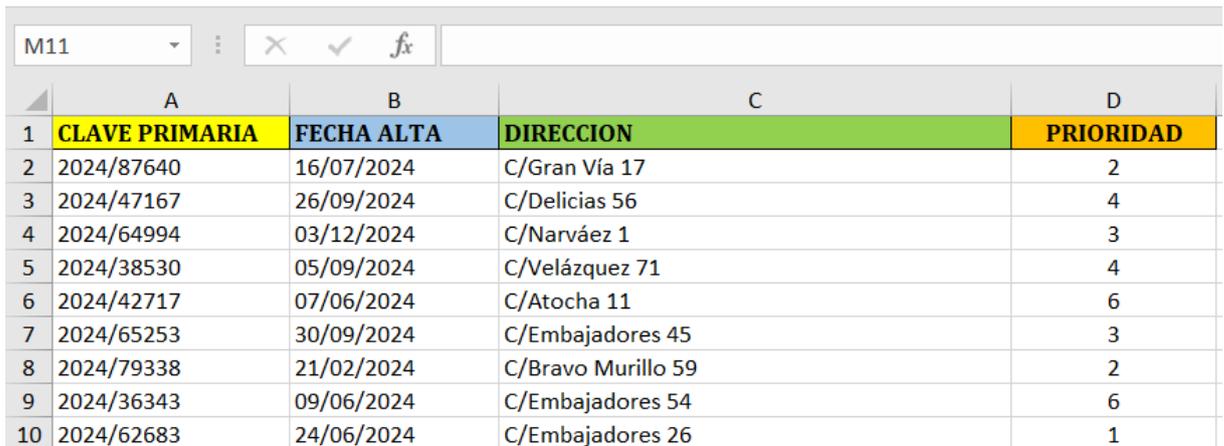
En este capítulo se explica el código de programación que se ha desarrollado para la optimización de rutas, la información de partida a facilitar por el usuario, las bibliotecas, clases y funciones utilizadas en el código, los pasos a seguir para determinar las rutas óptimas, y la información de salida que obtiene el usuario tras su ejecución.

### 3.1 Información de entrada.

Las empresas que utilizan el software, cuya actividad puede ser diferente, van dando de alta los emplazamientos a los que es necesario asistir, asignándoles un código que hace de clave primaria. La información de partida del usuario es la siguiente:

- **Clave primaria (primary key):** identifica de forma única las altas generadas. Hay que tener en cuenta que la empresa puede necesitar, por ejemplo, resolver varias incidencias en el mismo emplazamiento, que se han podido generar en distinta fecha, probablemente con prioridad de atención diferente.
- **Fecha de alta:** especifica el día, mes y año del alta.
- **Dirección:** tipo de vía, nombre y número al que se requiere ir. Para mayor simplicidad del código se ha considerado que pertenecen a Madrid.
- **Prioridad:** se corresponde con un número entero de 0 a 9 que valora la relevancia del alta generada, siendo 0 la máxima y 9 la mínima. Cada empresa establece su criterio en base al cual asignar una u otra prioridad. A modo de ejemplo, para una empresa que se dedica al mantenimiento de ascensores, un atrapamiento tendría prioridad 0, variaciones de velocidad podría tener prioridad 1, mal funcionamiento de las puertas prioridad 2, etc.

A partir de esta información se va actualizando diariamente un archivo Excel con las altas que se van generando, que debe tener el formato que se muestra en la figura siguiente y guardarse con el nombre “reparto.xlsx”.



	A	B	C	D
1	<b>CLAVE PRIMARIA</b>	<b>FECHA ALTA</b>	<b>DIRECCION</b>	<b>PRIORIDAD</b>
2	2024/87640	16/07/2024	C/Gran Vía 17	2
3	2024/47167	26/09/2024	C/Delicias 56	4
4	2024/64994	03/12/2024	C/Narváez 1	3
5	2024/38530	05/09/2024	C/Velázquez 71	4
6	2024/42717	07/06/2024	C/Atocha 11	6
7	2024/65253	30/09/2024	C/Embajadores 45	3
8	2024/79338	21/02/2024	C/Bravo Murillo 59	2
9	2024/36343	09/06/2024	C/Embajadores 54	6
10	2024/62683	24/06/2024	C/Embajadores 26	1

Fig. 2.- Formato del archivo de Excel "reparto.xlsx" con la información de entrada a facilitar por el usuario. Fuente de elaboración propia.

### 3.2 Justificación del entorno de programación seleccionado.

En el panorama actual del desarrollo de software científico y de optimización, Python se distingue por ofrecer un conjunto de características que lo convierten en una herramienta excepcional para abordar problemas complejos como la planificación de rutas.

Desde una perspectiva computacional, Python cuenta con numerosas bibliotecas que permiten realizar operaciones complejas con un buen rendimiento. Aunque lenguajes como C++ pueden ofrecer un rendimiento ligeramente superior, Python compensa esta pequeña diferencia con una capacidad de desarrollo significativamente más rápida y una curva de aprendizaje mucho más suave.

Otra ventaja de Python es la facilidad de desarrollo, que se fundamenta en una sintaxis clara, concisa y legible que permite a los desarrolladores expresar conceptos complejos en pocas líneas de código. A diferencia de otros lenguajes como Java o C++, que requieren una estructura más compleja y múltiples líneas para tareas simples, Python promueve un estilo de programación directo y natural.

El soporte de la comunidad y la abundancia de recursos son otros aspectos que distinguen a Python.

La comunidad de soporte de Python representa uno de sus activos más valiosos. Se trata de una comunidad global, diversa y extremadamente activa que contribuye constantemente al ecosistema del lenguaje.

La documentación de Python se distingue por su calidad y exhaustividad. Cada herramienta (bibliotecas y frameworks especializados) cuenta con documentación detallada, ejemplos de uso, mejores prácticas y guías de implementación.

Finalmente, aunque se escapa del alcance del código desarrollado el ecosistema de machine learning de Python representa otra ventaja fundamental. Las librerías como scikit-learn, TensorFlow y PyTorch facilitan la implementación de algoritmos

inteligentes que pueden mejorar continuamente los procesos de optimización de rutas. Esta capacidad de integrar técnicas de inteligencia artificial y aprendizaje automático sitúa a Python en una posición privilegiada frente a otros lenguajes más tradicionales o especializados.

En conclusión, Python ofrece el equilibrio perfecto entre rendimiento computacional, facilidad de desarrollo, herramientas y flexibilidad. Es por esto que resulta un lenguaje de programación muy adecuado para crear un software de optimización de rutas.

### 3.3 Bibliotecas de Python utilizadas.

Una biblioteca en Python es un conjunto de módulos y paquetes que proporcionan funciones, clases y herramientas predefinidas que facilitan la programación. Las bibliotecas permiten realizar tareas específicas sin necesidad de escribir todo el código desde cero.

Las bibliotecas son útiles porque ahorran tiempo y esfuerzo, pues permiten reutilizar código ya probado y optimizado, evitando la necesidad de implementarlo manualmente y facilitando la resolución de problemas complejos.

#### 3.3.1 Biblioteca estándar de Python.

La biblioteca estándar de Python es muy amplia, y ofrece una gran cantidad de producciones. La biblioteca contiene módulos incorporados (escritos en C) que brindan acceso a las funcionalidades del sistema como entrada y salida de archivos que serían de otra forma inaccesibles para los programadores en Python, así como módulos escritos en Python que proveen soluciones estandarizadas para los diversos problemas que pueden ocurrir en el día a día en la programación.

A continuación, se explican algunas de las funciones que se han utilizado en el código desarrollado:

- **input ()**

La función `input()` permite interactuar con el usuario solicitándole una entrada desde el teclado. La función muestra un mensaje opcional (el "prompt") que le dice al usuario qué debe escribir, y luego devuelve el texto ingresado como una cadena (tipo `str`).

```
# Pido por teclado al usuario del programa los emplazamientos a cubrir en coche planificadas para cada empleado a lo largo de su jornada laboral.
```

```
emp_coche_trabajador = input ('¿A cuántos emplazamientos va a ir cada trabajador en coche?').strip()
```

El método `strip()` se utiliza para eliminar todos los espacios en blanco iniciales y finales de una cadena.

- **int ()**

La función `int()` devuelve un objeto entero construido a partir de un número o una cadena, o devuelve 0 si no se proporcionan argumentos.

```
# Convierto a entero el número introducido por teclado (como string)
emp_coche_trabajador = int(emp_coche_trabajador)
```

- **len (s)**

La función `len ()` devuelve el tamaño (el número de elementos) de un objeto. El argumento puede ser una secuencia (como una cadena, un objeto `byte`, una tupla, lista o rango) o una colección (como un diccionario, un `set` o un *frozen set*).

```
# Recorro las filas del DataFrame andando_df desde la segunda fila (la enumeración comienza en cero) , ya que la primera contiene los nombres de los campos hasta la última, que se corresponde con len(andando_df)
for j in range(1,len(andando_df)):
```

```
    # Con iloc[j] voy tomando el elemento j del campo DIRECCION
    direccion_destino=andando_df['DIRECCION'].iloc[j]
```

- **round (number, ndigits=None)**

La función `round` retorna `number` redondeado a `ndigits` de precisión después del punto decimal. Si `ndigits` es omitido o es `None`, retorna el entero más cercano a su entrada.

```
# Reodondeo al segundo decimal el float km_distancia_coche
km_distancia_coche=round(km_distancia_coche,2)
```

- **min (arg1, arg2, \*args, key=None) y max(arg1, arg2, \*args, key=None)**

Las funciones `min` y `max` devuelven respectivamente el menor o mayor elemento en un iterable o de dos o más argumentos.

```
# Determino los tiempos mínimo y máximo de la lista tiempos_desde_primer_punto, que contiene los tiempos de trayecto desde el punto origen.
min_tiempo=min(tiempos_desde_primer_punto)
max_tiempo=max(tiempos_desde_primer_punto)
```

- **zip ()**

La función `zip` retorna un iterador de tuplas, donde la tupla *i*-ésima contiene el elemento *i*-ésimo de cada uno de los argumentos iterables.

```
# datos_trayecto_transporte_publico es una lista que almacena tuplas. Cada tupla contiene dos valores, la distancia en km y el tiempo en minutos para un trayecto en transporte público.
datos_trayecto_transporte_publico.append((km_distancia_tp,min_tiempo_tp))
# Separo la lista en dos listas individuales. El operador * desempaqueta los elementos de la lista y zip(*...) reagrupa los primeros elementos de todas las tuplas en una lista y los segundos elementos en otra.
lista_distancia_transporte_publico,lista_tiempo_transporte_publico =
zip(*datos_trayecto_transporte_publico)
```

### 3.3.2 Pandas

Pandas es una biblioteca de manipulación de datos en Python para datos tabulares. Es decir, datos en forma de filas y columnas, también conocidos como DataFrames.

Para leer un archivo Excel y convertir su contenido en un DataFrame se utiliza la *función* `read_excel`. Por ejemplo, el código que permite la lectura del Excel de partida es:

```
import pandas as pd
excel_direcciones = 'reparto.xlsx'
df = pd.read_excel(excel_direcciones,sheet_name=0)
```

Una clase en Python es una estructura de programación que permite definir un conjunto de métodos y atributos que describen un objeto o entidad. La *clase* `ExcelWriter` permite escribir DataFrames en hojas de Excel. Esta clase se utiliza principalmente cuando desea guardar varias hojas y adjuntarlas.

La *función* `to_excel` permite escribir un objeto en una hoja de Excel.

Para escribir un solo objeto en un archivo `.xlsx` de Excel, solo es necesario especificar un nombre de archivo de destino. Para escribir en varias hojas, es necesario crear un objeto `ExcelWriter` con un nombre de archivo de destino y especificar una hoja en el archivo para escribir.

El código para escribir el Excel con el reparto que se ha generado, especificando en hojas diferenciadas los emplazamientos a los que se irá en transporte público y aquellos para los que es necesario disponer de coche se muestra a continuación.

```
with pd.ExcelWriter('Reparto_generado.xlsx') as writer:
    andando_df_reparto.to_excel(writer,sheet_name='Paquetes_trabajo_andando',index=False)
    coche_df_reparto.to_excel(writer,sheet_name='Paquetes_trabajo_coche',index=False)
```

La *función* `iloc` permite seleccionar datos de un DataFrame o Series utilizando índices enteros. Es una herramienta poderosa para trabajar con ubicaciones específicas en términos de posición, similar al acceso mediante índices en listas o matrices en Python.

```
# Con iloc[1:] selecciono todas las filas desde la posición 1 hasta el final del DataFrame
andando_df_peos, excluyendo la fila en la posición 0.
andando_df_pesos=andando_df_pesos.iloc[1:]
```

La *función* `concat` concatena una cantidad arbitraria de objetos, Series, o DataFrames a lo largo de un eje. Al concatenar DataFrames con ejes con nombre, pandas intentará preservar estos nombres de índice/columna siempre que sea posible.

```
# Concateno el DataFrame "fila", que se corresponde con la fila [index - 1] del DataFrame
"andando_df_pesos" con el DataFrame "andando_df_reparto"
fila=andando_df_pesos.iloc[[index-1]]
andando_df_reparto=pd.concat([andando_df_reparto,fila],ignore_index=True)
```

La *función* `drop` elimina filas o columnas especificando nombres de etiquetas y ejes correspondientes, o especificando directamente nombres de índice o columna.

```
#Elimino del dataframe coche_df las columnas DISTANCIA_TRANSP_PUBLICO y TIEMPO_TRANSP_PUBLICO
```

```
coche_df=coche_df.drop(columns=['DISTANCIA_TRANSP_PUBLICO'])  
coche_df=coche_df.drop(columns=['TIEMPO_TRANSP_PUBLICO'])
```

La *función dropna* elimina de un DataFrame las filas en las que falta al menos un elemento y la *función tolist* convierte una columna del DataFrame en una lista.

```
#Creo una lista a partir de la columna 'DIRECCION' del DataFrame df, eliminando previamente  
filas en blanco (evito así posibles errores)
```

```
direcciones = df['DIRECCION'].dropna().tolist()
```

La *función sort\_values()* se utiliza para ordenar las filas de un DataFrame o Series en función de los valores de una o más columnas (o índices, en el caso de Series).

```
#Ordeno por prioridad y dentro de esta por días transcurridos desde alta y distancia, ascending  
= True de menor a mayor, y ascending = False de mayor a menor.  
andando_df=andando_df.sort_values(by=['PRIORIDAD','DIFERENCIA_DIAS','DISTANCIA_  
TRANSP_PUBLICO'],ascending=[True,False,True])
```

### 3.3.3 Googlemaps

La *biblioteca googlemaps* permite interactuar con las APIs de Google Maps e integrar las funcionalidades de Google Maps en aplicaciones de Python, facilitando tareas como geocodificación, cálculo de rutas, distancias entre ubicaciones, búsqueda de lugares, y más. Es especialmente útil para desarrolladores que necesitan automatizar consultas relacionadas con ubicaciones o procesar datos geoespaciales.

Con *googlemaps*, se pueden realizar operaciones como convertir direcciones en coordenadas geográficas (latitud y longitud) o viceversa, calcular rutas óptimas para el transporte (en automóvil, bicicleta, transporte público, etc.), obtener estimaciones de tiempos y distancias de viaje, y buscar puntos de interés cercanos.

Para interactuar con las diversas APIs de Google Maps, como Directions, Geocoding, Distance Matrix o Places entre otras, se requiere utilizar la *clase Client*.

### 3.3.4 Datetime

La *biblioteca datetime* proporciona clases para manipular fechas y horas. Si bien la implementación permite operaciones aritméticas con fechas y horas, su principal objetivo es poder extraer campos de forma eficiente para su posterior manipulación o formateo.

De esta biblioteca necesitamos importar la *clase datetime*.

El objeto *datetime* es un único objeto que contiene toda la información de un objeto *date* y un objeto *time*.

El *objeto date* representa una fecha (año, mes y día) en un calendario idealizado, el calendario gregoriano actual se extiende indefinidamente en ambas direcciones.

Un *objeto time* representa una hora (local) del día, independiente de cualquier día en particular. Los objetos de tiempo admiten comparaciones de igualdad y orden, donde 'a' se considera menor que 'b' cuando 'a' precede a 'b' en el tiempo.

Para convertir una cadena a un objeto datetime, utilizamos la *función.strptime* de la clase datetime.datetime. El primer argumento de la función es la cadena de caracteres que contine la fecha y el segundo argumento especifica el formato de la cadena de entrada. Seguidamente se explica parte del código desarrollado, en el que se hace uso de la biblioteca datetime y de sus funciones.

# Importo la biblioteca datetime y la clase datetime de esta biblioteca

```
import datetime as dt
from datetime import datetime
```

# Mediante la función today() determino la fecha actual, que se almacena en la variable current\_date.

```
current_date = dt.datetime.today()
```

# Creo una lista en la que se van a almacenar los días transcurridos desde las fechas de alta datos\_dias\_diferencia = []

```
for fecha in fecha_alta:
```

```
    # Almaceno como objeto datetime la fecha del DataFrame (cadena de caracteres)
```

```
    date_object = dt.datetime.strptime(fecha,"%d/%m/%Y")
```

```
    # La función abs me permite determinar la diferencia entre dos tiempos
```

```
    diferencia_fechas = abs(current_date - date_object)
```

```
    # La función days permite extraer los días
```

```
    dias=diferencia_fechas.days
```

```
    # Con la función append se van agregando elementos (días transcurridos) al final de la lista
```

```
    datos_dias_diferencia.append(dias)
```

### 3.4 Interfaces de programación de aplicaciones (APIs) empleadas.

Una **API (Interfaz de Programación de Aplicaciones)** es un conjunto de reglas y protocolos que permite a diferentes aplicaciones o sistemas interactuar entre sí de manera estructurada y eficiente. En términos simples, una API actúa como un puente que facilita la comunicación entre dos componentes de software, permitiendo que compartan datos o funcionalidades sin necesidad de conocer los detalles internos de su implementación. Este enfoque modular promueve la interoperabilidad, la escalabilidad y el desarrollo ágil de aplicaciones al permitir que los desarrolladores se centren en integrar componentes preexistentes en lugar de construirlos desde cero.

Las APIs son fundamentales en la arquitectura moderna de software y abarcan un amplio espectro de aplicaciones, desde servicios web hasta hardware. Un ejemplo típico es el uso de APIs de terceros, como Google Maps o Stripe, que permiten a los desarrolladores incorporar mapas interactivos o procesar pagos en sus aplicaciones sin necesidad de desarrollar esas capacidades desde cero.

En resumen, una API no solo facilita la integración y el intercambio de datos, sino que también permite que las aplicaciones evolucionen de manera más rápida y eficiente al aprovechar tecnologías y servicios ya existentes.

### 3.4.1 API Directions de GoogleMaps

Existen múltiples APIs en el mercado que permiten determinar la distancia entre dos direcciones y los tiempos de trayecto, pero finalmente se optó por utilizar la API Directions de GoogleMaps

La API de Directions es un servicio que acepta una solicitud HTTP y muestra instrucciones sobre cómo llegar entre ubicaciones en formato JSON o XML. Con la API de Directions, se pueden obtener instrucciones sobre cómo llegar a un destino en varios medios de transporte, como transporte público, vehículos particulares, a pie o en bicicleta.

La API de Directions muestra las rutas más eficientes cuando se calculan las instrucciones sobre cómo llegar. El producto considera los siguientes elementos de viaje cuando determina la ruta más eficiente:

- Tiempo de viaje (principal)
- Distancia
- Cantidad de giros

**Google Cloud** ofrece una **prueba** en la que el usuario puede usar sin cargo un valor de hasta USD 300. La prueba se aplica a la **primera cuenta de facturación** que se crea.

Durante la prueba, la primera cuenta de Facturación de Cloud del usuario está en modo de prueba y no se realizan cargos a su forma de pago. La prueba finaliza cuando termina el período de prueba de 90 días o cuando se agota el valor de USD 300, lo que ocurra primero.

Google Maps Platform cuenta además con un crédito mensual recurrente de \$200 (consulta Precios para Maps, Routes y Places). El crédito mensual se aplica a cada cuenta de Facturación de Cloud relacionada con Maps que se crea.

El funcionamiento es el siguiente: durante la prueba, los cargos se deducen primero del crédito mensual recurrente de USD 200 de Google Maps Platform y si los cargos superan los USD 200 en un mes determinado, el importe excedente se deduce del saldo restante de los USD 300 asignados a la prueba de Google Cloud Platform.

Como ya se mencionó, antes de que finalice la prueba, se debe actualizar la primera cuenta de Facturación de Cloud a una cuenta pagada. Una vez que se realice la actualización, **el crédito mensual de USD 200 se seguirá aplicando a esta primera cuenta de Facturación de Cloud, incluso después de que finalice la prueba.**

El plan de facturación de la API Directions se muestra a continuación:

RANGO DE VOLUMEN MENSUAL (precio por CONSULTA)		
0-100,000	100,001-500,000	Más de 500,000
USD 0.005 por cada una (USD 5.00 cada 1,000)	USD 0.004 por cada uno (USD 4.00 cada 1,000)	Para obtener información sobre los precios por volumen, <a href="#">comunicate con el equipo de Ventas</a> <a href="#">↗</a> .

Fig. 3.- Plan de pricing de la API Directions de Google Maps. Fuente Google. "Uso y facturación de la API de Directions" Google Maps Platform Documentation.

Con este plan de pricing podrían realizarse 40.000 consultas mensuales de forma gratuita.

Las principales razones para utilizar finalmente esta API, además de la posibilidad de consultas gratuitas, fueron las siguientes:

- **Datos precisos y actualizados:** la API Directions utiliza datos de tráfico en tiempo real, lo que permite calcular tiempos de trayecto considerando las condiciones actuales, como congestión, cierres de calles o retrasos inesperados. Esto es crucial para optimizar rutas y tomar decisiones basadas en información actualizada.
- **Capacidad de personalización:** permite especificar múltiples parámetros como modo de transporte (automóvil, bicicleta, caminata, transporte público), preferencias de ruta (evitar peajes, autopistas o tráfico) y restricciones horarias. Esto facilita la creación de rutas adaptadas a las necesidades del usuario o del sistema.
- **Integración con otros servicios de Google Maps:** la API Directions se complementa con otras APIs de Google, como Geocoding (para convertir direcciones a coordenadas), Distance Matrix (para calcular múltiples tiempos de trayecto simultáneamente), y Places (para localizar puntos de interés). Estas integraciones hacen que el sistema sea más robusto y versátil.
- **Facilidad de integración y soporte técnico:** Google ofrece una excelente documentación y soporte para la API, lo que facilita su implementación y resolución de problemas. Además, la API cuenta con SDKs (software development kit) para diversas plataformas y lenguajes de programación.
- **Optimización en tiempo real:** gracias a las capacidades de monitoreo continuo del tráfico y la flexibilidad de ajustes dinámicos, se puede optimizar rutas en tiempo real, permitiendo a los usuarios reaccionar rápidamente ante imprevistos.

Por último, se explica el código que permite realizar las consultas mediante esta API:

```
# Importo la biblioteca googlemaps
import googlemaps
```

```
#En la variable Maps almaceno la instancia para acceder a las APIs, que se efectúa haciendo
uso de la clase Client de la biblioteca googlemaps y facilitando el API Key del usuario.
Maps = googlemaps.Client(key= 'AlzaSyAMH1O_VN9JjGr5W_CyBPX5ufSrhIN1LDw')
```

```
# Realizo una consulta a la API directions y almaceno la información devuelta en la variable
distancia_transporte_público (el nombre no es muy apropiado porque la consulta devuelve
mucha más información). Se facilitan las direcciones de origen, de destino y el modo en que
se hace el recorrido (en este caso 'transit' ya que se hace en transporte público). Otros modos
son 'walking' y 'driving'.
```

```
distancia_transporte_publico = Maps.directions(direccion_oficina,direccion_destino + ',
Madrid, Spain',mode='transit')
```

```
# Accedo al valor de la distancia (la devuelve en metros) y la convierto a kilómetros.
km_distancia_tp = (distancia_transporte_publico[0]['legs'][0]['distance']['value']) /1000
```

```
# Accedo al valor de la distancia (la devuelve en metros) y la convierto a kilómetros.
min_tiempo_tp = (distancia_transporte_publico[0]['legs'][0]['duration']['value']) /60
```

Para saber dónde se encuentra la distancia y el tiempo de trayecto es conveniente formatear y mostrar en la consola el contenido de la respuesta de la API de una manera ordenada y fácil de leer.

El método `json.dumps` de la biblioteca estándar `json` de Python convierte un objeto de Python en una cadena de texto con formato JSON. El parámetro `indent` indica el número de espacios que se usarán para la sangría en cada nivel del JSON, haciendo que los elementos del JSON se alineen jerárquicamente. Por otra parte, al establecer `ensure_ascii=False`, se permite que los caracteres no ASCII (como letras con acentos, ñ, etc.) se muestren directamente en el JSON, facilitando su lectura.

```
# Importo la biblioteca json
import json
# Muestro la respuesta en formato JSON ordenado
print(json.dumps(distancia_transporte_público, indent=4, ensure_ascii=False))
```

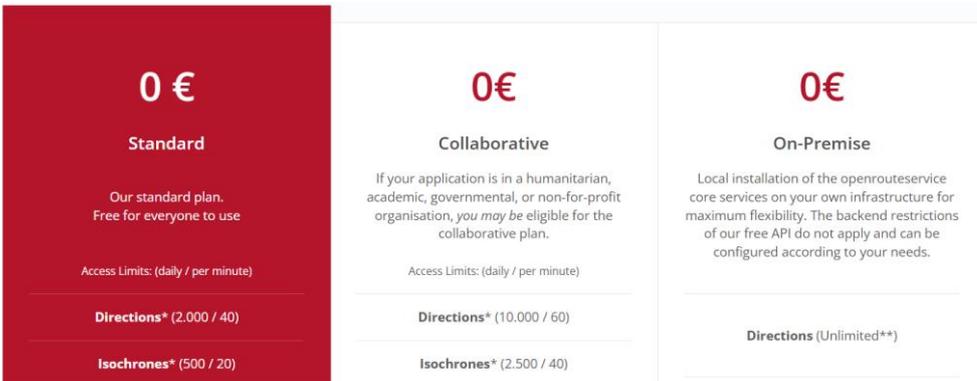
### 3.4.2 Otra alternativa, APIs `opencagedata` y `openrouteservice`

Las APIs `OpenRouteService` y `OpenCageData` son herramientas que ofrecen funcionalidades relacionadas con mapas, geocodificación y enrutamiento, similares a las de Google Maps, pero con características y enfoques específicos que las distinguen.

`OpenCageData` es una API especializada en geocodificación y geocodificación inversa que permite traducir datos de direcciones y coordenadas de forma precisa, proporcionando contexto geográfico.

`OpenRouteService` es una API que proporciona diversas funciones entre las que se incluyen cálculo de rutas, isócronas, análisis geoespacial y geocodificación inversa. Ambas APIs ofrecen ventajas clave frente a Google Maps, como ser totalmente gratuitas, transparencia al utilizar datos abiertos (basados en `OpenStreetMap`), no tener restricciones de uso comercial (ideal para proyectos que buscan independencia) y políticas más claras sobre el manejo de datos de usuario en comparación con gigantes tecnológicos.

La API `Openrouteservice` tiene límites de consulta diarios y por minuto para los planes `Standard` y `Collaborative`, retrasando esto mucho el tiempo de ejecución del código porque se hace necesario utilizar la función `time.sleep()` para retrasar la ejecución del código. Los distintos planes se muestran a continuación:



Plan	Costo	Descripción	Límites de acceso (diario / por minuto)
Standard	0 €	Our standard plan. Free for everyone to use.	Directions* (2.000 / 40) Isochrones* (500 / 20)
Collaborative	0 €	If your application is in a humanitarian, academic, governmental, or non-for-profit organisation, you may be eligible for the collaborative plan.	Directions* (10.000 / 60) Isochrones* (2.500 / 40)
On-Premise	0 €	Local installation of the openrouteservice core services on your own infrastructure for maximum flexibility. The backend restrictions of our free API do not apply and can be configured according to your needs.	Directions (Unlimited**)

Fig. 4.- Planes que ofrece la API `Openrouteservice`, todos ellos gratuitos. Fuente: `OpenRouteService "Plans and Pricing" OpenRouteService`.

La API Opencagedata permite una prueba gratuita con 2.500 consultas diarias. El plan de pricing es el siguiente:

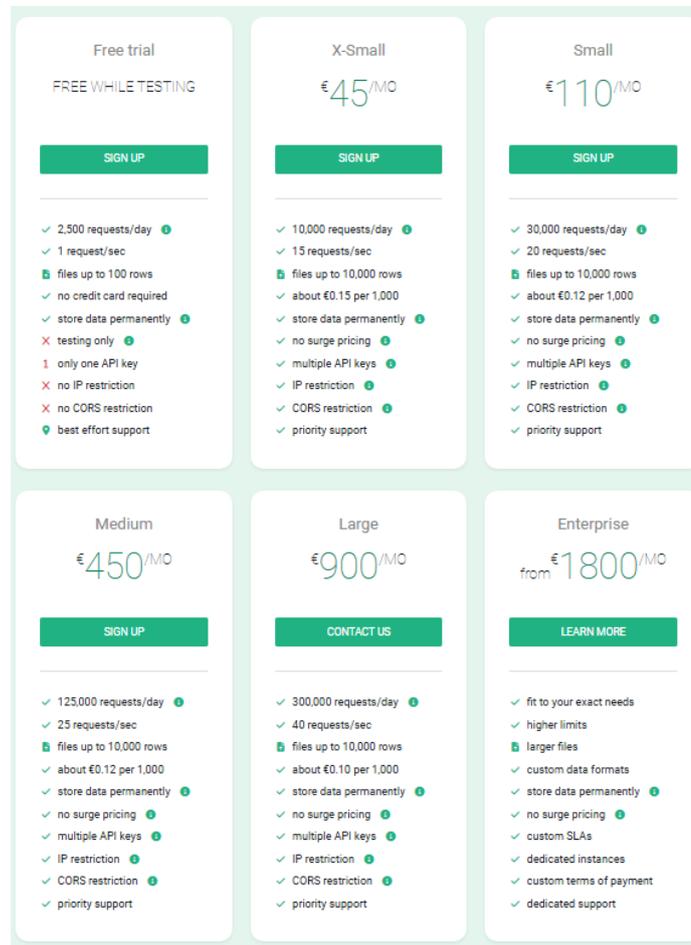


Fig. 5.- Planes que ofrece Opencage, entre cuyas APIs se encuentra Opencagedata. Fuente: OpenCage. "Pricing" OpenCage.

Para la utilización de ambas APIs es necesario importar la *biblioteca openrouteservice*.

```
# Importo la biblioteca openrouteservice
import openrouteservice
```

El código con el que se determinarían las coordenadas de las direcciones de origen y destino con la API OpenCageData sería el siguiente:

```
# clave api de opencagedata (obtener coordenadas de una dirección)
api_key_opencagedata = '2c8b0b7e8efa43dc88a442398a165ddf'
direccion = 'C/Bustamante 16, Madrid, Spain'
# Construyo la URL para hacer una solicitud HTTP a la API de OpenCageData
```

# La f que se antepone a una cadena en Python indica que es una f-string o string formateada. Las f-strings permiten incluir expresiones o valores de variables directamente dentro de una cadena utilizando llaves {}

```
url=f"https://api.opencagedata.com/geocode/v1/json?q={direccion}&key={api_key_opencage  
data}"
```

#Con requests.get envío una solicitud GET a la URL generada, obteniendo una respuesta de la API. Con .json convierto la respuesta en formato JSON para que pueda procesarse fácilmente.

```
response = requests.get(url).json()
```

# Accedo a la latitud y a la longitud y las almaceno en las respectivas variables

```
latitud = response['results'][0]['geometry']['lat']
```

```
longitud = response['results'][0]['geometry']['lng']
```

# Creo un vector con las coordenadas de la dirección

```
coordenadas = (latitud, longitud)
```

Una vez se han determinado las coordenadas, mediante la API OpenRouteService se obtiene fácilmente la distancia entre las direcciones origen y destino. El código se muestra a continuación:

# clave api de openrouteservice (cálculo de rutas a pie y en coche)

```
api_key_openrouteservice='5b3ce3597851110001cf6248829748ab2565482cbeb869218015  
db39'
```

# Construyo la URL para hacer una solicitud HTTP a la API de OpenRouteService.

Hay que especificar `foot_walking` si el recorrido se realiza andando o `driving_car` si fuese en coche.

```
url_andando = f"https://api.openrouteservice.org/v2/directions/foot-  
walking?api_key={api_key_openrouteservice}&start={origen[1]},{origen[0]}&end={destino[1]},  
{destino[0]}"
```

#Con requests.get envío una solicitud GET a la URL generada, obteniendo una respuesta de la API.

```
response_andando = requests.get(url_andando).json()
```

# Accedo a la distancia y la almaceno en la variable `distancia_andando`. Se divide por 1.000 para pasar a km.

```
distancia_andando =
```

```
response_andando['features'][0]['properties']['segments'][0]['distance']/1000
```

## 3.5 Estructura del código.

### 3.5.1 Importación de bibliotecas y otras configuraciones iniciales:

Se importa las bibliotecas necesarias para trabajar con los distintos tipos de datos y con las APIs de GoogleMaps. Es conveniente importar, además de las bibliotecas `pandas`, `datetime` y `googlemaps`, a las que ya se hizo referencia en el apartado 4.3, las bibliotecas `requests` y `json`.

La *biblioteca requests* permite enviar solicitudes HTTP, por ejemplo a la API web `Openrouteservice`.

La *biblioteca json* sirve para trabajar con datos en formato JSON, que es el formato típico de las respuestas de APIs.

A continuación, se establecen las claves API del usuario.

Para obtener una clave API de Google Maps se necesita crear una cuenta de Google Cloud y un proyecto. Dentro del proyecto, seleccionando APIs y servicios > Biblioteca, se busca Directions API, que es la API que se utiliza en el código, y se habilita. En el menú de la izquierda, se selecciona APIs y servicios > Credenciales, se hace clic en el botón Crear credenciales y se selecciona Clave de API, generará así la API key, que aparecerá en la pantalla. Es importante proteger tu clave para evitar abusos. Por último, Google Maps requiere que configures un método de pago, aunque muchos servicios tienen un uso gratuito limitado tal y como ya se ha explicado anteriormente.

Para obtener las API key de Opencagedata y de Openrouteservice hace falta registrarse con una cuenta de correo electrónico. El plan de prueba de Opencagedata y el plan standard de Openrouteservice no requieren especificar cuenta ni tarjeta bancaria.

### 3.5.2 Almacenamiento de la información de entrada. Cálculo de distancias y tiempos en transporte público y generación de un archivo .xlsx con esta información.

Se lee el Excel de partida direcciones\_reparto.xlsx y se almacena la información contenida en el mismo en un DataFrame nombrado como df, del que se extraen las columnas que contienen las direcciones y las fechas de alta.

Seguidamente, se guarda en una variable la dirección de la oficina, punto de salida de los trabajadores para asistir a los distintos emplazamientos, así como la fecha actual (en la que se ejecuta el código).

Mediante un primer bucle for que recorre la lista con las direcciones, haciendo uso de la API directions de GoogleMaps, se van guardando los datos de los trayectos en transporte público (distancia en km y tiempo en min) desde la oficina a los distintos emplazamientos.

Con un segundo bucle for que recorre la lista con las fechas de alta se calculan y se almacenan los días que han transcurrido desde que se dio de alta el emplazamiento por ser necesario asistir al mismo por personal de la empresa.

Al DataFrame que se obtuvo a partir del Excel reparto.xlsx se añaden tres columnas que contienen respectivamente los días transcurridos desde el alta, la distancia y el tiempo en transporte público a los diferentes emplazamientos. El nuevo DataFrame así generado se guarda como un archivo Excel con nombre archivo reparto\_con\_datos\_transporte\_publico.xlsx.

La información facilitada por el archivo reparto\_con\_datos\_transporte\_publico.xlsx se muestra en la siguiente figura:

	A	B	C	D	E	F	G
1	CLAVE PRIMARIA	FECHA ALTA	DIRECCIÓN	PRIORIDAD	DIFERENCIA_DIAS	DISTANCIA_TRANSP_PUBLICO	TIEMPO_TRANSP_PUBLICO
2	2024/87640	16/07/2024	C/Gran Vía 17	2	165	3,75	28,52
3	2024/47167	26/09/2024	C/Delicias 56	4	93	0,46	6,93
4	2024/64994	03/12/2024	C/Narváez 1	3	25	3,75	26,53
5	2024/38530	05/09/2024	C/Velázquez 71	4	114	4,65	24,15
6	2024/42717	07/06/2024	C/Atocha 11	6	204	3,73	28,37
7	2024/65253	30/09/2024	C/Embajadores 45	3	89	2,15	22,08
8	2024/79338	21/02/2024	C/Bravo Murillo 59	2	311	6,4	36,23

Fig. 6.- Archivo reparto\_con\_datos\_transporte\_público.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx. Fuente: elaboración propia, generado a partir de la ejecución del código.

Se ha considerado interesante generar este archivo Excel para poder valorar la opción de asistir a alguno de los emplazamientos en transporte público, a pesar de que a criterio del código se establezca como necesario ir en coche. Puede ser interesante, por ejemplo, en caso de que la disponibilidad de coches de la empresa sea menor que la habitual, o incluso para hacer asignaciones adicionales a empleados en puntos próximos a la empresa en caso de que terminasen su trabajo programado para ese día cuando todavía falte tiempo para que terminen su jornada laboral.

### 3.5.3 Determinación de los emplazamientos a los que se puede asistir en transporte público y de aquellos en los que se utilizará el coche.

A partir del DataFrame obtenido genero dos nuevos DataFrames, `andando_df` y `coche_df`, que contendrán las filas para las que el tiempo en transporte público sean menor o mayor o igual que 25 minutos respectivamente. Diferencio así los emplazamientos a los cuales se asistirá en transporte público y aquellos para los que se recurrirá al coche.

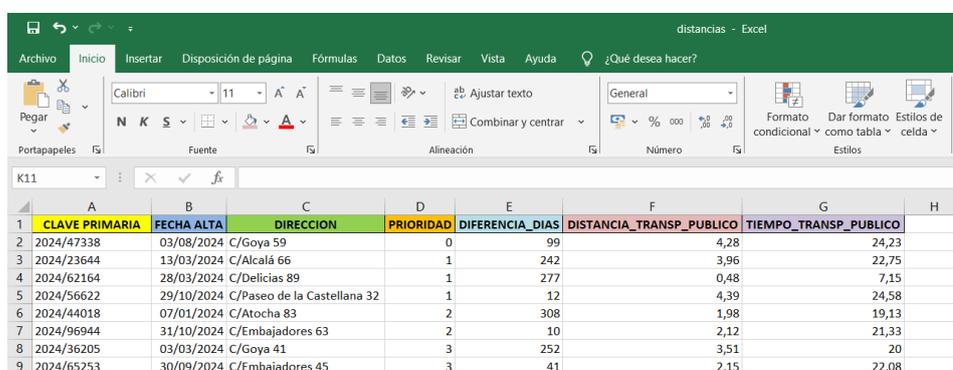
En el DataFrame `coche_df` se eliminan las columnas `DISTANCIA_TRANSP_PUBLICO` y `TIEMPO_TRANSP_PUBLICO`. Posteriormente se calcularán las distancias y tiempos, pero en coche.

Las filas del DataFrame `andando_df` se ordenan atendiendo a los criterios de prioridad (más prioritarias primero), días transcurridos (primero aquellas para las que han transcurridos más días desde el alta) y distancias (las más próximas antes).

Se crea una lista con las direcciones del DataFrame `coche_df`. Esta lista se recorre con un bucle `for` y haciendo uso de la API `directions` de GoogleMaps con el modo `driving` se generan dos listas con las distancias y tiempos en coche. Al DataFrame `coche_df` se añaden las columnas `DISTANCIA_COCHE` y `TIEMPO_COCHE` con el contenido de estas listas.

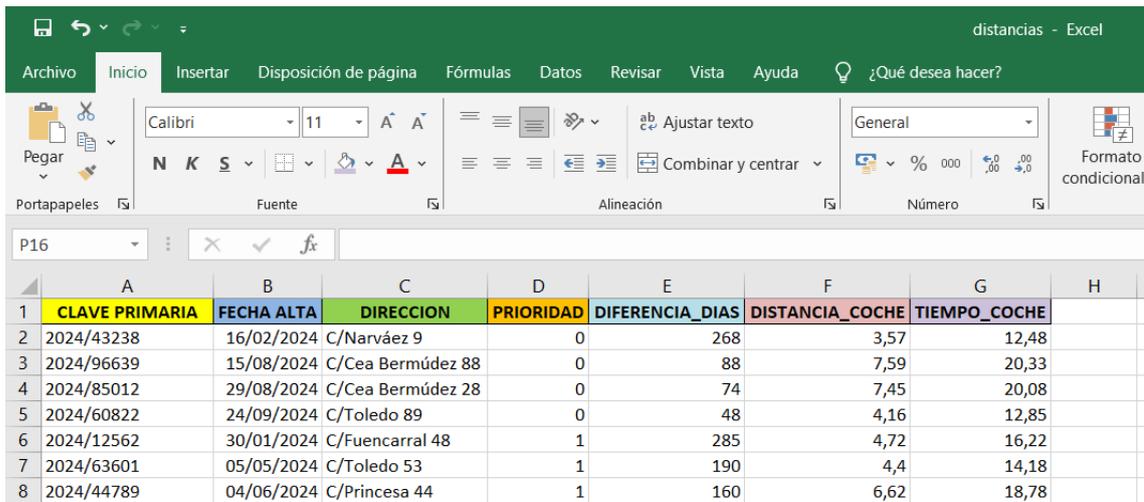
Las filas del DataFrame `coche_df` se ordenan de forma análoga a las de `andando_df`, aunque de modo que figuren primero las direcciones más alejadas respecto de la oficina.

Por último, genero un archivo Excel "distancias.xlsx" con dos hojas que contienen las altas que se van a cubrir en transporte público y en coche, con la información de distancias y tiempos desde la oficina. Los campos de las hojas "Andando" y "Coche" se muestran en la siguiente figura:



	A	B	C	D	E	F	G	H
1	CLAVE PRIMARIA	FECHA ALTA	DIRECCION	PRIORIDAD	DIFERENCIA DIAS	DISTANCIA_TRANSP_PUBLICO	TIEMPO_TRANSP_PUBLICO	
2	2024/47338	03/08/2024	C/Goya 59	0	99	4,28	24,23	
3	2024/23644	13/03/2024	C/Alcalá 66	1	242	3,96	22,75	
4	2024/62164	28/03/2024	C/Delicias 89	1	277	0,48	7,15	
5	2024/56622	29/10/2024	C/Paseo de la Castellana 32	1	12	4,39	24,58	
6	2024/44018	07/01/2024	C/Atocha 83	2	308	1,98	19,13	
7	2024/96944	31/10/2024	C/Embajadores 63	2	10	2,12	21,33	
8	2024/36205	03/03/2024	C/Goya 41	3	252	3,51	20	
9	2024/65253	30/09/2024	C/Embajadores 45	3	41	2,15	22,08	

Fig. 7.- Hoja Andando del archivo `distancias.xlsx` generado al ejecutar el código a partir del archivo `reparto.xlsx`. Fuente: elaboración propia, generado a partir de la ejecución del código.



	A	B	C	D	E	F	G	H
1	CLAVE PRIMARIA	FECHA ALTA	DIRECCION	PRIORIDAD	DIFERENCIA_DIAS	DISTANCIA_COCHE	TIEMPO_COCHE	
2	2024/43238	16/02/2024	C/Narváez 9	0	268	3,57	12,48	
3	2024/96639	15/08/2024	C/Cea Bermúdez 88	0	88	7,59	20,33	
4	2024/85012	29/08/2024	C/Cea Bermúdez 28	0	74	7,45	20,08	
5	2024/60822	24/09/2024	C/Toledo 89	0	48	4,16	12,85	
6	2024/12562	30/01/2024	C/Fuencarral 48	1	285	4,72	16,22	
7	2024/63601	05/05/2024	C/Toledo 53	1	190	4,4	14,18	
8	2024/44789	04/06/2024	C/Princesa 44	1	160	6,62	18,78	

Fig. 8.- Hoja Coche el archivo distancias.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx. Fuente: elaboración propia, generado a partir de la ejecución del código.

### 3.5.4 Cálculo de las asignaciones de emplazamientos que se cubrirán en transporte público y en coche.

Primeramente, se pregunta al usuario si se desean programar visitas andando a los emplazamientos, ya que podría ser que la empresa decidiese cubrir antes los emplazamientos más alejados para que no se acumulen demasiados o en vista de que vaya a tener que reparar algún vehículo próximamente. Seguidamente, en caso de respuesta afirmativa, se pregunta cuántos empleados van a cubrir emplazamientos andando (paquetes de trabajo andando) y a cuántos emplazamientos va a asistir cada empleado a lo largo de su jornada laboral.

Mediante un **primer bucle for**, se van a ir calculando los paquetes de trabajo.

La dirección de la primera visita se toma la primera del DataFrame andando\_df, caracterizada por su clave primaria.

**Anidados con este primer bucle for** se requieren otros **tres bucles for** para obtener los paquetes.

Haciendo uso de un **segundo bucle for** se realizan consultas a la API para determinar distancias y tiempos al resto de direcciones del DataFrame andando\_df. Las distancias y tiempos se almacenan en listas.

El orden en que se realizan las demás asignaciones se efectúa en función del valor que tome un factor de peso total, que tiene en cuenta la prioridad, días transcurridos desde la fecha de alta y las distancias en transporte público respecto del primer emplazamiento al que asiste el trabajador. Para facilitar el cálculo se crea un DataFrame andando\_df\_pesos a partir del DataFrame andando\_df, que contendrá, además de las claves primarias, fechas de alta, direcciones, prioridades y diferencia entre fecha de alta y actual, las distancias y tiempos determinados por la API, y los pesos desde la dirección (tres pesos parciales denominados peso\_prioridad, peso\_dia y peso\_tiempo, que toman valores entre 1 y 10, y el peso total).

Previamente a calcular los pesos parciales es necesario conocer los valores mínimo y máximo de los días transcurridos desde las fechas de alta, que se obtienen con las funciones `min` y `max` a partir de una lista generada a partir de la columna `DIFERENCIA_DIAS` del DataFrame `andando_df_pesos`. Igualmente, se obtienen los valores mínimo y máximo de la lista con los tiempos desde origen a los distintos emplazamientos.

Los cálculos de los pesos parciales se determinan haciendo uso de un **tercer bucle for** planteando la ecuación de una recta. Así, para el `peso_día` se sabe que para el valor mínimo de días transcurridos el peso parcial es 1 y para el máximo es 10, de modo que a partir de estos dos puntos, si en la iteración `z` del bucle los días transcurridos son `días_transcurridos_z`, el peso parcial viene dado por:

$$\text{peso\_dia} = 10 - (9/(\text{max\_dias} - \text{min\_dias})) * (\text{días\_transcurridos}[z] - \text{min\_dias})$$

Procediendo de forma similar se calculan `peso_tiempo` y `peso_prioridad`.

El peso total se determina sumando los tres pesos parciales, de modo que varía entre 3 y 30.

$$\text{peso\_total} = \text{peso\_prioridad} + \text{peso\_dia} + \text{peso\_tiempo}$$

La fórmula para calcular el peso total podría modificarse, por ejemplo, multiplicando por una constante numérica algún peso parcial. Aunque esto permitiría una mayor personalización al usuario también añade una complejidad adicional, ya que es probable que el usuario no sepa determinar unas constantes que resulten apropiadas. Se ha optado por la simplicidad, no permitiéndose su modificación.

Una vez recorrido este bucle se completan las columnas `PESO_TIEMPO`, `PESO_PRIORIDAD`, `PESO_DIAS_TRANSCURRIDOS` y `PESO_TOTAL` del DataFrame `andando_df_pesos` con los pesos calculados durante la ejecución de este bucle `for`. Finalmente se ordenan las filas de mayor a menor peso total.

Un **último bucle for** permite genera un DataFrame con los paquetes de trabajo `andando`, que en el código se ha nombrado como `andando_df_reparto`.

Como los tiempos y distancias de la primera fila de este DataFrame han de ser desde la dirección de la oficina, se toma la primera fila del DataFrame `andando_df` y se añaden los

factores de peso parciales con valor 10 y peso total 30 (tras haberse ordenado las filas de `andando_df` se asume que la primera es la más prioritaria).

Las siguientes filas, que contendrán las visitas restantes, son las primeras del DataFrame `andando_df_pesos` hasta completar el paquete.

Las asignaciones que se van realizando en los distintos paquetes, se localizan a partir de la clave primaria en el DataFrame `andando_df` y se eliminan para evitar así volver a programarlas cuando se ejecute una nueva iteración del primer bucle para generar un nuevo paquete. Igualmente se van eliminando del DataFrame `df` de partida que contiene todas las altas.

Todo este código va embebido dentro de un bucle `while` que se ejecuta de forma continua mientras el usuario desee generar nuevos grupos de paquetes de visitas `andando`.

Para los emplazamientos a cubrir en coche el código es muy similar, valorándose que no merece la pena entrar en su explicación.

### **3.5.5 Obtención de archivos Excel con el reparto generado y con las visitas por realizar.**

Por último, se generan dos archivos Excel, “Reparto\_generado.xlsx” y “Direcciones\_final.xlsx”.

El archivo “Reparto\_generado.xlsx”, tiene dos hojas,

“Paquetes\_trabajo\_andando” y “Paquetes\_trabajo\_coche” en la que se vuelva la información de los DataFrames `andando_df_reparto` y `coche_df_reparto`.

El archivo “Direcciones\_final.xlsx” se genera a partir del DataFrame `df`, en el que ya se borraron las filas con las visitas que se han programado.

## **3.6 Información de salida facilitada tras la ejecución.**

En los apartados anteriores ya se han explicado los archivos `reparto_con_datos_transporte_publico.xlsx` y `distancias.xlsx` obtenidos tras la ejecución a partir del archivo `reparto.xlsx` facilitado por el usuario. El primero de ellos contiene las distancias y tiempos en transporte público desde la empresa hasta los distintos emplazamientos, mientras que el segundo determina los emplazamientos para los que es razonable utilizar el transporte público y aquellos para los que se recomienda emplear el coche, facilitando igualmente las distancias y tiempos.

Sin embargo, la información de salida más relevante son los archivos `Reparto_generado.xlsx` y `Direcciones_final.xlsx`.

El archivo `Reparto_generado.xlsx` recoge los paquetes de trabajo propuestos para los distintos empleados, especificado por parte del usuario cuántos paquetes en transporte público y en coche se quieren generar y el número de emplazamientos que desea que contenga cada paquete. Adicionalmente, especifica los días que han transcurrido desde la fecha de alta, distancias y tiempos, así como los factores de peso asociados. Este archivo

contiene dos hojas, `Andando` y `Coche`, con los paquetes a cubrir en transporte público y en coche respectivamente.

En las figuras siguientes, fig. 9 y fig. 10, se muestra el archivo `Reparto_generado.xlsx` generado al ejecutar el código a partir del archivo `Reparto.xlsx` facilitado por el usuario.

Durante la ejecución se han pedido tres paquetes de trabajo a realizar en transporte público con tres emplazamientos cada uno y un único paquete a realizar en coche con cinco emplazamientos. Como puede observarse, los pesos totales de los emplazamientos de cada paquete son cada vez menores, ya que se consideran primero los emplazamientos más prioritarios. La distancia del primer emplazamiento de cada paquete se calcula respecto a la oficina, mientras que las siguientes se determinan respecto del primer emplazamiento (son pequeñas porque se busca que sean próximas).

	A	B	C	D	E	F	G	H	I	J	K
1	CLAVE PRIMARIA	FECHA ALTA	DIRECCION	PRIORIDAD	DIFERENCIA_DIAS	DISTANCIA_TRANSP_PUBLICO	TIEMPO_TRANSP_PUBLICO	PESO_TIEMPO	PESO_PRIORIDAD	PESO_DIAS_TRANSCURRIDOS	PESO_TOTAL
2	2024/47338	03/08/2024	C/Goya 59	0	156	4,28	24,23	10,00	10	10,00	30,00
3	2024/74104	20/11/2024	C/Velázquez 43	4	47	0,56	8,10	9,13	6	9,20	24,33
4	2024/56622	29/10/2024	C/Paseo de la Castellana 32	1	69	1,61	16,22	6,13	9	8,64	23,77
5											
6	2024/23644	13/03/2024	C/Alcalá 66	1	299	3,96	22,75	10,00	10	10,00	30,00
7	2024/38530	05/09/2024	C/Velázquez 71	4	123	1,29	9,43	10,00	7	7,25	24,25
8	2024/88519	21/12/2024	C/Paseo de la Castellana 20	5	16	1,59	15,65	7,13	5	10,00	22,13
9											
10	2024/62164	28/03/2024	C/Delicias 89	1	284	0,48	7,15	10,00	10	10,00	30,00
11	2024/47167	26/09/2024	C/Delicias 56	4	102	0,28	3,95	10,00	7	8,95	25,95
12	2024/96944	31/10/2024	C/Embajadores 63	2	67	0,84	11,62	5,48	8	10,00	23,48
13	2024/72418	16/09/2024	C/Atocha 64	6	112	1,35	11,42	5,60	4	8,65	18,24

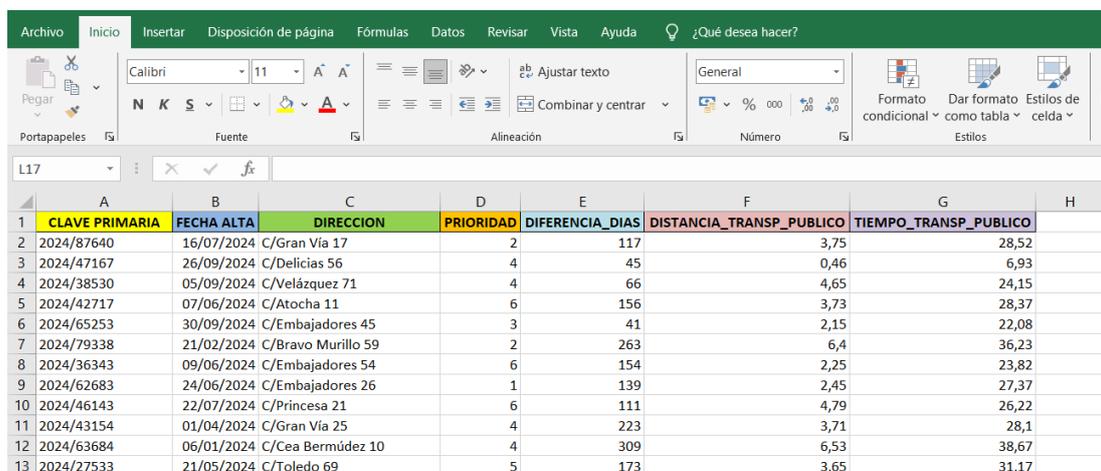
Fig. 9.- Hoja "Andando" del archivo Reparto\_generado.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx tras especificar el número de paquetes y emplazamientos por paquete. Fuente: elaboración propia, generado a partir de la ejecución del código.

	A	B	C	D	E	F	G	H	I	J	K
1	CLAVE PRIMARIA	FECHA ALTA	DIRECCION	PRIORIDAD	DIFERENCIA_DIAS	DISTANCIA_COCHE	TIEMPO_COCHE	PESO_TIEMPO	PESO_PRIORIDAD	PESO_DIAS_TRANSCURRIDOS	PESO_TOTAL
2	2024/43238	16/02/2024	C/Narvéez 9	0	325	3,57	12,48	10,00	10	10,00	30,00
3	2024/64994	03/12/2024	C/Narvéez 1	3	34	0,15	0,85	9,93	7	9,87	26,80
4	2024/19519	24/03/2024	C/Narvéez 3	3	288	0,13	0,73	10,00	8	3,06	21,06
5	2024/89512	21/05/2024	C/Goya 69	3	230	0,73	3,33	8,55	7	4,62	20,16
6	2024/87640	16/07/2024	C/Gran Vía 17	2	174	2,37	8,42	5,70	8	6,12	19,81
7											
8											

Fig. 10.- Hoja "Coche" del archivo Reparto\_generado.xlsx generado al ejecutar el código a partir del archivo reparto.xlsx tras especificar el número de paquetes y emplazamientos por paquete. Fuente: elaboración propia, generado a partir de la ejecución del código.

Finalmente, el usuario debe saber qué emplazamientos faltan por asignar, información que se recoge en el archivo Direcciones\_final.xlsx. Este archivo debe actualizarlo la empresa incorporando los nuevos emplazamientos a los que es necesario asistir, y una vez actualizado contendría la información de entrada que habría que facilitar en la próxima ejecución del código para generar los nuevos paquetes de trabajo.

Sería necesario renombrar el archivo como Reparto.xlsx en la nueva ejecución y eliminar las columnas DIFERENCIA\_DIAS, DISTANCIA\_TRANSP\_PUBLICO y TIEMPO\_TRANSP\_PUBLICO.



	A	B	C	D	E	F	G	H
1	CLAVE PRIMARIA	FECHA ALTA	DIRECCION	PRIORIDAD	DIFERENCIA_DIAS	DISTANCIA_TRANSP_PUBLICO	TIEMPO_TRANSP_PUBLICO	
2	2024/87640	16/07/2024	C/Gran Vía 17	2	117		3,75	28,52
3	2024/47167	26/09/2024	C/Delicias 56	4	45		0,46	6,93
4	2024/38530	05/09/2024	C/Velázquez 71	4	66		4,65	24,15
5	2024/42717	07/06/2024	C/Atocha 11	6	156		3,73	28,37
6	2024/65253	30/09/2024	C/Embajadores 45	3	41		2,15	22,08
7	2024/79338	21/02/2024	C/Bravo Murillo 59	2	263		6,4	36,23
8	2024/36343	09/06/2024	C/Embajadores 54	6	154		2,25	23,82
9	2024/62683	24/06/2024	C/Embajadores 26	1	139		2,45	27,37
10	2024/46143	22/07/2024	C/Princesa 21	6	111		4,79	26,22
11	2024/43154	01/04/2024	C/Gran Vía 25	4	223		3,71	28,1
12	2024/63684	06/01/2024	C/Cea Bermúdez 10	4	309		6,53	38,67
13	2024/27533	21/05/2024	C/Toledo 69	5	173		3,65	31,17

Fig. 11.- Archivo Reparto\_generado.xlsx, en el que, tras la ejecución, determinados los paquetes de trabajo, se recogen los emplazamientos que han quedado sin asignar. Fuente: elaboración propia, generado a partir de la ejecución del código.

## Capítulo 4. Plan de negocio.

### 4.1 Oportunidad.

#### 4.1.1 Formulación de la oportunidad.

En España, el sector servicios representa el 63,3% del total de empresas activas, lo que equivale a unas 2.060.614 empresas según los datos del Directorio Central de Empresas (DIRCE) publicados por el INE para enero de 2024. Aproximadamente el 8,5% de estas empresas están vinculadas a actividades de logística, almacenamiento y transporte. Por otra parte, las empresas dedicadas a servicios técnicos y mantenimiento suponen también un porcentaje significativo, aunque los subsectores específicos no están completamente desglosados en los datos recientes del INE. Haciendo una estimación muy conservadora podría cifrarse el número de empresas nacionales dedicados a esta actividad en un 10% del total de empresas del sector servicios, es decir, aproximadamente 200.000 empresas.

En la Comunidad de Madrid, donde se centrarán las campañas de marketing para empezar a captar clientes, según datos del INE y la Cámara de Comercio de Madrid hay aproximadamente 500,000 empresas activas. De estas, al menos un 15%-20% están en sectores relacionados con transporte, logística, distribución o servicios que requieren vehículos. Esto equivale a unas 75,000-100,000 empresas en Madrid y su área metropolitana.

En términos generales, se puede estimar que alrededor de un 20%-30% de las empresas de estos sectores estarían interesadas en el software desarrollado, dependiendo de sus necesidades operativas y factores como su nivel de madurez tecnológica o su compromiso ambiental. Pueden considerarse por lo tanto 10,000-15,000 empresas potenciales en Madrid y alrededores, que podrían considerar el software como una solución interesante para sus operaciones.

Las empresas que necesitan realizar estas actividades suelen recurrir a la subcontratación y al outsourcing para centrarse en su actividad principal, reducir gastos fijos, tener mayor flexibilidad, ser más eficientes (se recurre a empresas especializadas) y transferir riesgo.

Las grandes empresas que desarrollen estas actividades ofrecen servicios integrales y cobertura geográfica amplia y cuentan con muchos recursos. Su número no es muy elevado, cabiendo destacar empresas como DHL, Seur, ISS Facility Services o MRW.

Las pequeñas empresas especializadas se centran en nichos específicos o mercados locales, ofrecen servicios más personalizados y tienen estructuras de costes más ligeras.

Volviendo de nuevo al capítulo introductorio del trabajo, las pequeñas y medianas empresas necesitan automatizar el proceso de determinación de las rutas para poder ser competitivos y buscan soluciones sencillas, que se aprendan a utilizar con facilidad.

Las nuevas empresas que se crean no suelen estar dispuestas a arriesgar demasiado, por lo que inicialmente prefieren realizar poca inversión en softwares de optimización de rutas.

Adicionalmente, los entornos urbanos padecen problemas de tráfico y de contaminación, existiendo una tendencia a la circulación de vehículos contaminantes en estas zonas, potenciándose alternativas más sostenibles como el transporte público, bicicleta, etc.

Un producto que contemple este tipo de alternativas supone un valor añadido significativo y es muy probable que a corto o medio plazo cada vez sean más las empresas que las contemplen en el ejercicio de su actividad.

#### 4.1.2 Descripción del mercado.

Según el informe de Mordor Intelligence “Análisis de participación y tamaño del mercado de software de optimización de rutas tendencias y pronósticos de crecimiento (2024-2029)” se estima que el tamaño del mercado de software de optimización de rutas es de 3.520 millones de dólares en 2023, habiendo mostrado un crecimiento significativo en los últimos años. Para 2029 se proyecta que alcance los 5.460 millones de dólares.

El tamaño del mercado varía según la fuente, el estudio de Infinitive Data Expert “Tamaño del mercado de software de optimización de rutas, análisis de la industria por implementación (local, en la nube): global, tendencias, participación y pronóstico 2023-20230”, el mercado se valora en 5.690 millones de dólares en 2022 y podría expandirse a una CAGR (tasa de crecimiento anual compuesto) del 15,1% entre 2023 y 2030.

El mercado está compuesto por una variedad de actores, desde grandes corporaciones hasta startups innovadoras. Entre los principales proveedores se encuentran Trimble Inc., Caliper Corporation, Descartes Systems Group Inc., ESRI Global Inc. y Google LLC.

La oferta se segmenta según el modo de implementación (en la nube o local), el tamaño de la organización (pequeñas y medianas empresas frente a grandes corporaciones) y la actividad desarrollada por los usuarios finales, que incluyen entrega de pedidos de alimentos, comercio minorista y bienes de consumo, servicios de campo, servicios de transporte y taxi, etc. Además, hay una segmentación clara según el modelo de transporte (flotas de vehículos propios, transporte público, movilidad compartida) y los criterios de optimización (costos, tiempos e impacto ambiental).

No obstante, el mercado comparte las siguientes características comunes:

- **Orientación a la eficiencia operativa:** todas las soluciones buscan reducir costos y mejorar la productividad optimizando recursos, ya sea tiempo, combustible o personal.
- **Uso de tecnología avanzada:** utilizan algoritmos sofisticados, inteligencia artificial, geolocalización y análisis de datos en tiempo real para ofrecer soluciones dinámicas y adaptativas.
- **Personalización y escalabilidad:** ofrecen soluciones adaptables a las necesidades específicas del sector y tamaños de empresas.
-

- **Integración** con sistemas existentes en las empresas como ERPs (Enterprise Resource Planning) o CRMs (Customer Relationship Management).

La propuesta de negocio puede considerarse una disrupción en un nicho específico del mercado que requiere la optimización de rutas para transporte con una estructura de necesidades heterogéneas, al integrar transporte público y automóviles particulares. Aunque no redefine el mercado global de optimización de rutas, sí plantea un enfoque novedoso y ofrece una propuesta de valor única frente a los softwares disponibles en el mercado actual. Supone una innovación incremental, ya que mejora las soluciones existentes al incorporar funcionalidades específicas, a la vez que cierta innovación disruptiva, al abordar necesidades particulares que no han sido plenamente satisfechas por las soluciones actuales.

Al enfocarse el producto en un segmento específico y relativamente desatendido, como la optimización de rutas para empleados que combina diferentes medios de transporte y prioridades, este nuevo producto tiene el potencial de crear un océano azul. Al ofrecer una propuesta de valor única, podría capturar a un segmento específico de clientes y diferenciarse de los competidores tradicionales, estableciéndose como líder en este nicho.

Este apartado se complementa por último con dos herramientas, el análisis DAFO, fundamental a la hora de desarrollar un plan estratégico, y el análisis de las 5 fuerzas de Porter, que permiten valorar capacidad del negocio para ser competitivo y rentable.

## Análisis DAFO



Fig. 12.- Análisis DAFO de la empresa que lanzará el software desarrollado. Fuente: Canva. "Diseño DAGa-GVaCy!" <https://www.canva.com> (plantilla).

## Análisis de las cinco fuerzas de Porter

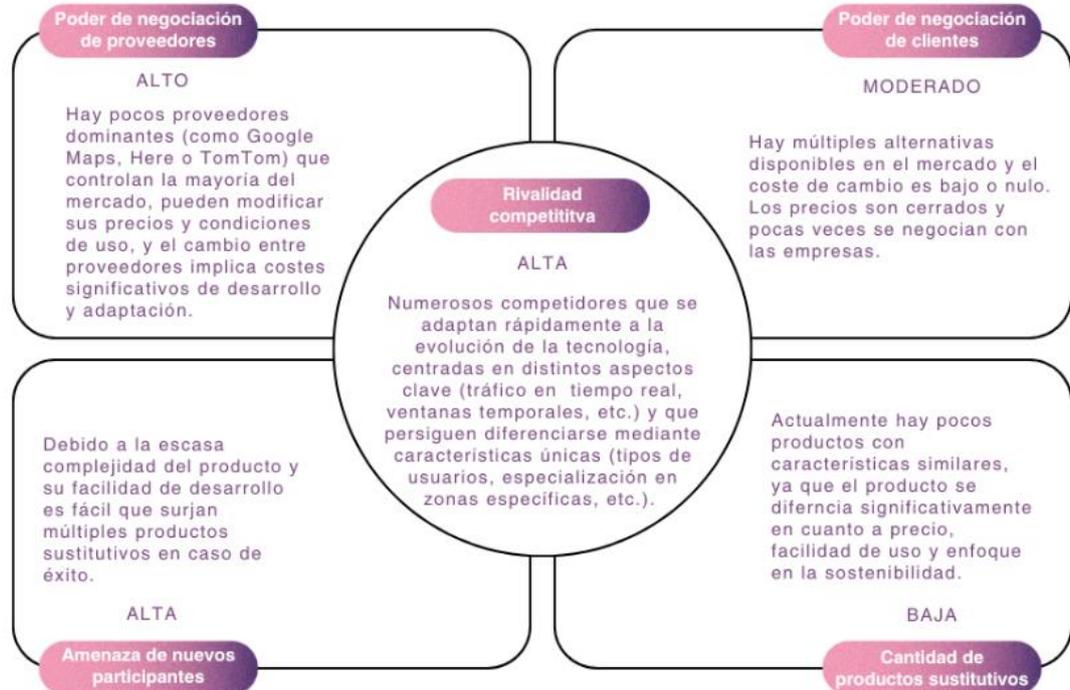


Fig. 13.- Análisis DAFO de la empresa que lanzará el software desarrollado. Fuente: Canva. "Diseño DAGa-OITzgy" <https://www.canva.com> (plantilla).

### 4.1.3 Modelo de negocio objetivo.

En este apartado se introduce el modelo de negocio que deseamos desarrollar, entendiendo modelo de negocio como la forma de generar beneficio económico sobre la oportunidad descrita. Un buen marco para describir el modelo de negocio y ampliamente utilizado en la empresa es el "business model canvas" desarrollado por Alex Ostelwolder.

A continuación, se presenta el "lienzo" del modelo de negocio y se describe a través del mismo el detalle de los diferentes componentes que serán desarrollados en los siguientes apartados.



Fig. 14.- Business model canvas de la empresa EcoRutas, que lanzará al mercado el software desarrollado. Fuente: elaboración propia.

## 4.2 Segmentación de clientes.

Un software de optimización de rutas de bajo precio, sencillo de usar y enfocado en trayectos en transporte público para zonas bien comunicadas, pero con limitaciones en la programación de un gran número de rutas, podría estar dirigido a segmentos de mercado con preferencia por herramientas accesibles y prácticas, con un enfoque en necesidades específicas y una tolerancia limitada a restricciones técnicas siempre que el software sea gratuito y fácil de usar. Caben destacar los siguientes segmentos.

- **Pequeñas empresas o microempresas locales:** negocios como repartidores independientes, empresas dedicadas a gestión de incidencias o reparaciones y empresas de mensajería de última milla o pequeños servicios de entrega que operen en áreas urbanas bien conectadas y no tengan una alta demanda de rutas complejas. Estas empresas valoran soluciones accesibles y de bajo costo para organizar sus desplazamientos diarios.
- **Trabajadores autónomos y freelancers:** profesionales como guías turísticos, agentes inmobiliarios o asesores que necesitan planificar visitas o recorridos en zonas cercanas y pueden aprovechar la cobertura de transporte público para desplazarse de forma eficiente sin incurrir en costos adicionales.
- **Organizaciones sin fines de lucro o comunitarias:** asociaciones vecinales, ONG o colectivos que trabajan en áreas urbanas y necesitan coordinar desplazamientos para actividades como eventos, campañas o entregas a pequeña escala, pero no cuentan con recursos para costear software avanzado.

### 4.3 Propuesta de valor.

#### 4.3.1 Atributos diferenciales de la propuesta de valor.

Los atributos preferentes del producto desarrollado para nuestros segmentos de mercado objetivo son la facilidad de uso, el precio y el enfoque en la sostenibilidad.

Las pequeñas empresas y trabajadores autónomos muchas veces carecen de conocimientos informáticos avanzados y valoran muy positivamente que el producto sea sencillo de utilizar e intuitivo. Por otra parte, no suelen tener unos beneficios muy altos por lo que no están dispuestos a pagar una gran cantidad de dinero por este servicio, aunque por un precio razonable si les resulta interesante. Además, gran cantidad de estas empresas operan en entornos urbanos en los que el transporte público puede ser una alternativa interesante ya que les permite además reducir costes, por lo que valoran muy positivamente que el producto contemple posibles rutas en transporte público.

Por otra parte, las organizaciones sin fines de lucro o comunitarias habitualmente disponen de pocos recursos y habitualmente están interesadas en la sostenibilidad.

A continuación, se realiza una evaluación de estos atributos para el producto que se va a lanzar al mercado:

- **Facilidad de uso:** manejar datos directamente desde un archivo Excel simplifica la implementación sin necesidad de configuraciones complejas o conocimientos técnicos avanzados.
- **Precio:** el software podría posicionarse como una opción más económica, ya sea con un pago único o una suscripción más accesible frente a sus competidores.
- **Enfoque en la sostenibilidad:** el producto ofrece la funcionalidad específica de identificar direcciones accesibles por transporte público. Al sugerir como modos de transporte el transporte público muestra un compromiso ambiental.

Para analizar los principales competidores del producto desarrollado, en primer lugar, se estudia el cuadrante mágico de Gartner de los softwares de optimización de rutas y el cuadrante en el que se encontraría.



Fig. 15.- Cuadrante mágico de Gartner de los softwares de optimización de rutas. Fuente: LanRoute. "Cuadrante Mágico de Gartner". <https://lanroute.com/>

El cuadrante recoge **4 divisiones**:

- **Líderes:** son los principales actores, se caracterizan por gran capacidad y habilidad para ejecutar. Son compañías modernas, que se anticipan y dan soluciones avanzadas. Son ServiceMax, ClickSoftware, Oracle e IFS.
- **Retadores o aspirantes:** tienen buenas soluciones, pero están centrados en un único aspecto del mercado. Son SAP y Salesforce.
- **Visionarios:** están cerca de los líderes por capacidad de anticipación, pero carecen de recursos para ejecutar sus planes. Son Microsoft y ServicePower.
- **Jugadores de nicho:** están centrados en nichos de mercado. Entre ellos se encuentra FieldAware, Comarch, Praxedo, Geoconcept, Accruent (Verisae), Astea International, CoreSystems u OverIT.

El software desarrollado estaría en el cuadrante de **jugadores de nicho** debido a que se enfoca en segmentos de mercado específicos, con preferencia por herramientas accesibles y prácticas, especialmente en entornos urbanos bien conectados con transporte público; está diseñado para ser sencillo de usar y económico, pero tiene limitaciones para ajustar parámetros de optimización o introducir nuevos; y aunque incluye factores diferenciadores carece de funcionalidades avanzadas características de líderes o visionarios.

El principal competidor de mi software actualmente es **Route4Me**, que se encuentra **entre los Visionarios o Jugadores de Nicho**, ya que ofrece soluciones accesibles para pymes con buena capacidad de personalización y facilidad de uso, aunque sin las capacidades de ejecución robustas de un líder.



*Fig. 16.- Principal competidor del producto desarrollado. Fuente: Route4Me. (n.d.). Planificación y optimización de rutas. <https://route4me.com>*

Route4Me se enfoca en ofrecer una solución accesible, sencilla de usar y adaptable a diferentes necesidades, características que también definen a mi software. Tanto mi producto como Route4Me están dirigidos a clientes que buscan herramientas económicas y prácticas, sin la necesidad de implementar sistemas complejos o costosos. Esta similitud en el público objetivo coloca a Route4Me como un rival directo.

Otra similitud que hace que Route4Me sea un competidor aún más directo es que también permite la carga de múltiples direcciones desde hojas de cálculo. Esta funcionalidad permite a los usuarios subir un archivo con direcciones de los destinos, y el software automáticamente genera rutas optimizadas considerando varios parámetros, como distancias, tiempo de conducción y restricciones específicas. Además, Route4Me tiene una interfaz intuitiva para cargar y mapear columnas (por ejemplo, asignar columnas de "Dirección", "Nombre del cliente" o "Tiempo estimado de visita"), lo que simplifica el proceso incluso para usuarios con poca experiencia técnica.

En lo que al precio se refiere, Route4Me comienza en 36€/mes. Mi producto se ofertará significativamente por debajo, estableciéndose el precio en el apartado 4.3.4. Modelo de precios e ingresos.

Route4Me ofrece múltiples funciones avanzadas como integraciones con sistemas externos, optimización multi-día y configuraciones específicas para diferentes industrias, estas características tienden a aumentar la curva de aprendizaje, especialmente para usuarios sin experiencia técnica o con necesidades logísticas básicas. El producto que yo he desarrollado no tiene ninguna de estas funciones por lo que tiene una curva de aprendizaje con una pendiente muy elevada, resultando extremadamente **sencillo** para el usuario.

Respecto al **servicio**, Route4Me ofrece opciones de soporte que incluyen asistencia telefónica y por correo electrónico. Mi empresa únicamente ofrece asistencia por correo electrónico, aunque se compromete a resolver las incidencias en menos de 24h.

En cuanto al grado de personalización, Route4Me permite personalizar las rutas en función de múltiples variables como tiempo de conducción, distancias, horarios de visita, restricciones de vehículos, y preferencias como rutas circulares o lineales. También ofrece herramientas específicas para personalizar rutas según necesidades más complejas, como evitar ciertas áreas o carreteras, agregar ventanas de tiempo para entregas y configurar prioridades de visitas. Además, funciona para diversos sectores como logística, servicios de campo, y entrega de última milla, con la flexibilidad de ajustar configuraciones para cada tipo de negocio. Comparativamente, la personalización de mi producto está muy limitada, pues no va más allá de la posibilidad de asignar prioridades a los emplazamientos según criterios determinados por el usuario.

En términos de sostenibilidad, Route4Me, a diferencia de mi producto, no se especializa en movilidad sostenible ni en transporte público. Esta diferenciación puede ser clave para atraer clientes para los cuales la sostenibilidad sea una de sus prioridades específicas.

A modo de resumen se muestra la curva de valor que compara el producto desarrollado con Route4Me.

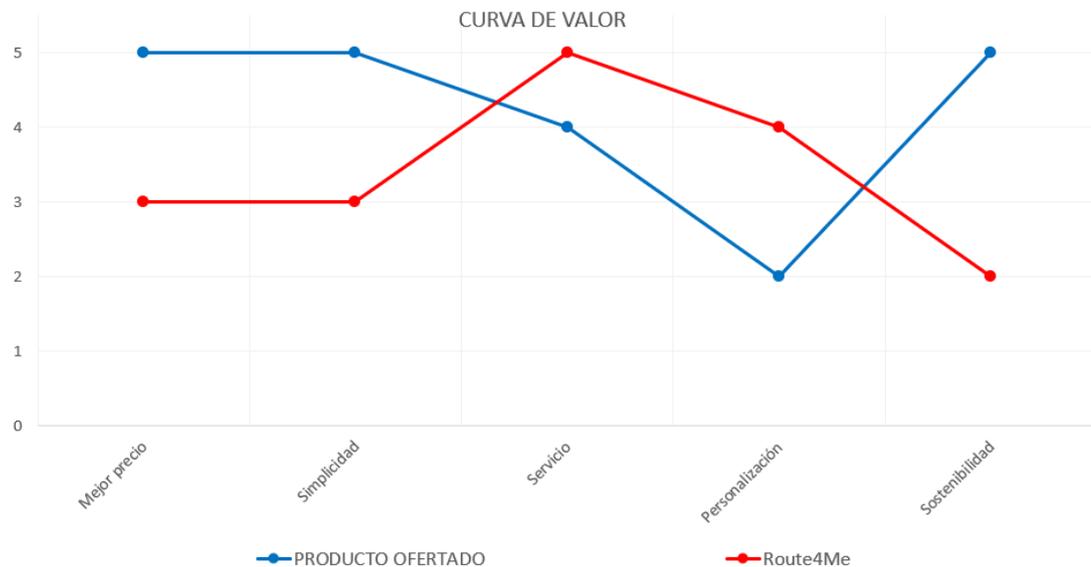


Fig. 17.- Curva de valor que compara las características del producto ofrecido con las de su principal competidor, Route4Me. Fuente: elaboración propia.

#### 4.3.2 Descripción de productos y servicios.

El producto que se ofrece, ya descrito de forma muy detallada en los apartados anteriores, consiste en un código de Python mediante el cual, a partir de un Excel con direcciones a las que tienen que asistir los empleados se pueden generar paquetes de trabajo diarios, siendo posibles el coche o el transporte público para zonas próximas a la oficina como modo de transporte. En la generación de los paquetes se tiene en cuenta no solo distancias, sino también prioridad y tiempo desde el cual surgió la necesidad de ir a cada emplazamiento, lo que supone una optimización de los paquetes generados.

Podría resultar interesante mejorar el código permitiendo incorporar como medio de transporte alternativo la bicicleta eléctrica, muy apropiada para zonas urbanas en las que esté limitado el acceso de vehículos diésel o gasolina y que podría mejorar tiempos respecto al transporte público debido a una baja frecuencia en determinadas franjas horarias. También, sería

atractivo para el usuario la posibilidad de añadir otros factores de peso a tener en cuenta para la optimización del reparto generado y de personalizar la fórmula del cálculo del peso total de los distintos emplazamientos. Por último, podría incorporarse un cálculo estimativo de la reducción de las emisiones de CO<sub>2</sub> por la empresa al optar por la utilización del transporte público.

Los beneficios que ofrece este producto a sus clientes son:

- **Tangibles e intangibles:**

- Ahorro de tiempo en la planificación de las rutas y en la asignación de los emplazamientos a los que debe ir cada empleado en su jornada laboral.
- Generación de los paquetes de trabajo de los empleados y de la relación de emplazamientos en archivos Excel, unificando formatos y evitando errores típicos de procedimientos manuales.
- Evita la desmotivación de los trabajadores por la realización de una tarea compleja, monótona y repetitiva.

- **Económicos y no económicos:**

- Reducción de costos operativos al optimizar rutas.
- Reduce la flota de vehículos necesaria y por ende el gasto asociado a la misma si se opta por el transporte público.
- Mejora en la satisfacción de empleados, que percibirá más fundamentados los traslados que realiza al ver los distintos pesos parciales y el peso total asociados a los emplazamientos que visita en su día a día.
- Los clientes estarán más satisfechos ya que la empresa realizará el trabajo en tiempos, desde que se genera el alta, que están ajustados a la prioridad.

- **Emocionales:**

- Mayor confianza en que las tareas de asignación de emplazamientos a cubrir por los empleados se cumplen de forma eficiente y sin retrasos.
- Tranquilidad para la empresa al manejar imprevistos de manera más ágil a partir de los archivos Excel generados mediante el código.
- Satisfacción con la empresa al incluir como beneficio social para sus empleados la tarjeta transporte.

- **Sociales:**

- Fomenta el uso del transporte público, apoyando la sostenibilidad.
- Reduce las emisiones de CO<sub>2</sub> de la empresa.

- Entre las incomodidades que evita el producto cabe destacar, la necesidad de planificar manualmente los emplazamientos a los que deben asistir los trabajadores, teniendo en cuenta múltiples factores como las prioridades, distancias o días transcurridos y la necesidad de conocer las rutas de transporte público para valorar si este medio de transporte es una buena opción.
- Este producto además elimina algunas inseguridades frecuentes en las empresas como la incertidumbre sobre si las asignaciones de rutas son eficientes, dudas acerca del cumplimiento de las tareas en los tiempos previstos y preocupaciones relacionadas con el impacto de paquetes de trabajo no optimizados.

#### 4.3.3 Plan de desarrollo y lanzamiento de productos y servicios.

El este apartado se presenta el cronograma de desarrollo y lanzamiento de nuevas versiones del producto. Paralelamente se justifican los gastos e inversiones asociados.

Para convertir un código de programación en una herramienta informática accesible, funcional y fácil de usar por parte del usuario requiere de una interfaz de usuario (UI), especialmente teniendo en cuenta que muchos de ellos no tienen conocimientos técnicos. El **mínimo producto viable (MPV)** consiste en el código de Python desarrollado, ya descrito ampliamente en el capítulo 3, junto con su interfaz. Tras el lanzamiento, el producto se irá mejorando y actualizando mediante nuevas versiones.

Lógicamente, alcanzar el mínimo producto viable y ofrecer nuevas versiones, requiere una inversión inicial y supone una serie de gastos, que se describen seguidamente:

- **Desarrollo de la interfaz de usuario:** se contratará a una empresa externa como Velneo, APPYWEB o Zaask. Para una aplicación con funcionalidades moderadas el proyecto puede suponer entre 3.000€ y 7.000€, considerándose que 4.500€ serán suficientes.
- **Trabajadores de los distintos departamentos de la empresa** (estructura de la empresa y costes que suponen los trabajadores en el apartado 4.8 Gestión de talento)
  - **Desarrolladores Python:** estará integrada por dos personas, siendo yo una de ellas. Trabajaré para alcanzar el MPV, mejorar el producto implementando nuevas funcionalidades y manteniéndolo actualizado. Además, hay que crear la página web y un knowledge base incluyendo FAQs.
  - **Ingenieros informáticos para dar soporte técnico:** entre sus funciones destaca la gestión de las incidencias de los clientes con el producto.
  - **Marketer digital:** juega un papel clave en el lanzamiento y en la promoción de las nuevas versiones, así como en el análisis de las posibles disconformidades y de las necesidades demandadas por parte de los segmentos de mercado objetivo. Mantendrá una posición muy próxima a los clientes y los ingenieros responsables de la gestión de incidencias, especialmente cuando se desarrollen nuevas versiones, responsabilizándose de analizar la información facilitada por estos.

- **Inversión en equipos informáticos** (se hace una estimación en el apartado 4.9. Sistemas de información).
- **Gastos operativos asociados a IT y comunicaciones** (contemplados en el apartado 4.9. Sistemas de información), **y a marketing** (recogidos en el apartado 4.3.4 Modelo de precios e ingresos).

Cuando se lance una nueva versión del producto, se enviará a los clientes un newsletter extraordinario informando de las novedades introducidas con la versión e informando de las demostraciones que se realizarán. Además, se actualizará el knowledge base.

Para abordar el desarrollo de la nueva versión se seguirá la **metodología scrum**, debido a su enfoque en la iteración mediante ciclos cortos, la colaboración con los clientes y la adaptabilidad, que minimizan el riesgo.

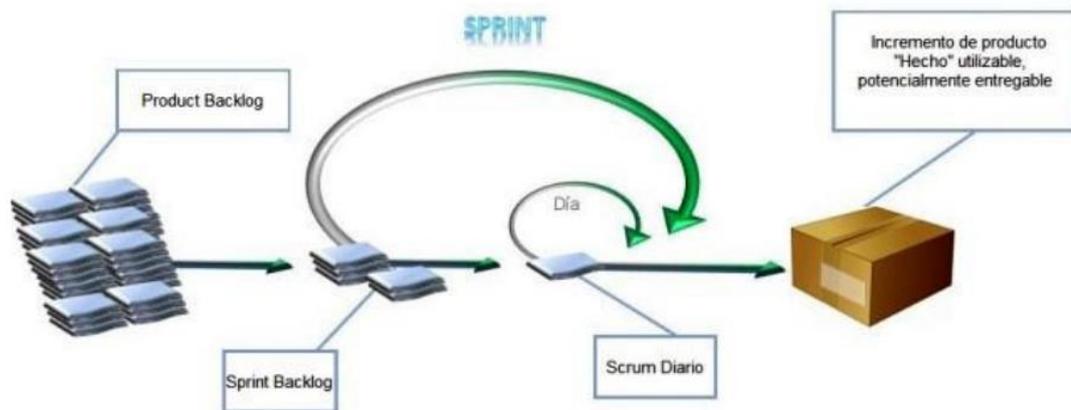


Fig. 18.- Resumen de la metodología scrum. Fuente: Lacort Peña, B. (2024-2025). Apuntes de la asignatura Dirección de grandes proyectos. Universidad Europea de Madrid.

En cuanto a los **roles del equipo**, yo asumiré el papel de *Scrum Master*, facilitando la metodología, eliminando impedimentos y asegurándome de que el equipo siga los principios ágiles (principios definidos en el Manifiesto Ágil). El *Development Team* estará integrado por los desarrolladores Python y por ingenieros informáticos encargados del soporte técnico, que contribuyen al equipo con su experiencia en seguridad informática y gestión de datos. El marketer digital desempeñará el papel de *Product Owner (PO)*, dado que tiene una visión cercana a las necesidades de los clientes y puede priorizar las características y correcciones según su impacto.

El primer paso será definir el *sprint backlog*, un listado ordenado y priorizado con los requisitos necesarios para la implementación de la nueva versión mediante el que se descompondrá el producto objetivo en tareas.

El desarrollo se dividirá en *sprints*, ciclos cortos y repetitivos de dos semanas de duración. Al inicio de cada sprint, se realizará la reunión de *Sprint Planning*, con una duración de 2h, para definir el objetivo del sprint, seleccionar los elementos del producto backlog y planificar cómo lograr el trabajo.

El desarrollo de una versión ampliamente aceptada por los clientes requiere la colaboración constante con ellos. Es por esto que se considera necesario realizar demostraciones

periódicas (*sprint reviews*) e incorporar encuestas para recoger comentarios directamente dentro de la herramienta (pueden utilizarse bibliotecas de encuestas como SurveyJS o Typeform Embed SDK).

Las demostraciones a los clientes se realizarán por videoconferencia (Zoom o Microsoft Teams) en una franja horaria fuera de la jornada laboral, por ejemplo, a las 20:00h, al final de cada sprint (cada 2 semanas) para asegurarme de que los clientes estén al tanto del progreso constante y puedan influir en el desarrollo de la herramienta. Un formato de demostración adecuado podría ser el siguiente:

- **Inicio (5-10 minutos):** se presentan brevemente los objetivos del sprint y lo que se logró.
- **Demostración (15-20 minutos):** se muestran las nuevas funcionalidades o cambios en la herramienta.
- **Interacción y Retroalimentación (15-20 minutos):** se permite que los clientes prueben las funcionalidades, hagan preguntas y expresen sus impresiones en tiempo real.
- **Cierre (5 minutos):** se resumen los comentarios recibidos y los próximos pasos.

Se integrarán encuestas breves colocando una pequeña ventana emergente (popup) que se activará después de que el cliente haya probado la nueva funcionalidad un determinado número de veces. Estas encuestas serán muy simples, realizándose las siguientes preguntas: ¿cuál es tu grado de satisfacción con esta nueva versión (escala de 1 a 5) ?, ¿qué mejorarías de esta versión? Se configurará una API para almacenar los datos de las encuestas en una base de datos SQL.

Las observaciones, sugerencias y puntos críticos destacados por los clientes se incorporarán al *Product Backlog* para priorizarlas en futuros sprints.

Las métricas como el porcentaje de errores críticos resueltos o el índice de satisfacción del cliente pueden ser indicadores clave del progreso y la aceptación de la versión.

Finalmente, se presenta el **cronograma**, mostrándose los tiempos que se estiman necesarios para terminar de desarrollar el MPV y lanzar el producto al mercado, así como para desarrollar las posibles nuevas versiones pensadas en este momento, aunque sujetas a cambios en función de la acogida y los intereses de los clientes.

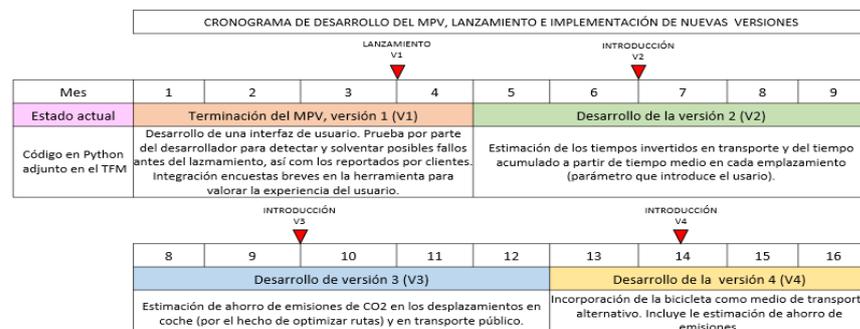


Fig. 19.- Cronograma de desarrollo del MPV, lanzamiento e implementación de nuevas versiones. Fuente: elaboración propia.

Como ya se ha comentado, para alcanzar el MPV se requiere la interfaz de usuario. También habrá que hacer una prueba del producto por parte del equipo de desarrolladores que permitirá detectar y solventar posibles fallos antes del lanzamiento. El tiempo que se estima necesario son cuatro semanas.

La versión 2 permitirá realizar una estimación de los tiempos invertidos en transporte y del tiempo acumulado a partir de tiempo medio en cada emplazamiento (parámetro que introduce el usuario). Antes de introducirla habrá que integrar encuestas breves en la herramienta para valorar la experiencia del usuario. Desarrollar la prueba de la versión 2 supondrá dos meses, dedicándose otros tres meses a alcanzar una versión final lo más satisfactoria posible para el conjunto de usuarios.

En la versión 3 se incorporará una estimación de ahorro de emisiones de CO<sub>2</sub> en los desplazamientos en coche (por el hecho de optimizar rutas) y en transporte público. Se valora que se necesitarán dos meses para crear una versión 3 de prueba y otros tres meses para alcanzar la versión final, ante las dudas de los requisitos que plantearán los usuarios.

Finalmente, se tiene pensado una versión 4 que plantee la bicicleta como medio de transporte alternativo, incluyéndose del mismo modo el ahorro de emisiones de CO<sub>2</sub> que supone. Implementar la prueba de la versión 4 llevará un mes y medio, considerándose necesarios otros dos meses y medio para alcanzar la versión final a partir del feedback de los clientes.

#### 4.3.4 Modelos de precios e ingresos.

Para fijar un precio al producto valoraré en este apartado los costes asociados al producto, el precio fijado por los principales productos de la competencia y la posible evolución de las ventas para un determinado precio, atendiendo a las características del producto y el valor percibido por parte de los potenciales clientes.

Los costes asociados se justifican a lo largo de los distintos apartados del capítulo 4 y su evolución a lo largo de los tres primeros años de la empresa (2026, 2027 y 2028) puede verse en el apartado 4.10.1 Flujos de caja.

Dentro de los costes fijos, el más representativo es el coste empresa de los trabajadores (alrededor de 100.000€ el primer año), seguido ya muy de lejos por los gastos operativos en IT y comunicaciones (no llegan a 9.000€).

Los costes variables vienen representados por las comisiones de las plataformas de venta, GitHub y GumRoad (5% y 10% del precio de venta respectivamente).

Tal y como se indicó en el apartado 4.3.1, el principal competidor, Route4Me ofrece planes que empiezan en 36€/mes, por lo que para que no resulte difícil capturar clientes, se decide diferenciarse en precio. Me gustaría que el precio fuese de 10€/mes, pero previamente en necesario estudiar si económicamente este precio resulta viable.

Un precio de 10 €/mes (casi 70 % más económico que Route4Me) es una **propuesta de valor agresiva** para empresas pequeñas y medianas que no pueden pagar tarifas más altas, por lo que muchas empresas que dudan en adoptar un software costoso como Route4Me podrían inclinarse hacia mi solución.

Por otra parte, se va a ofrecer un mes de prueba gratuito, lo que reduce significativamente las barreras de entrada. Al probar el software sin riesgo y confirmar su utilidad, es más probable que los usuarios se conviertan en clientes de pago.

Respecto a los clientes potenciales, si los primeros años la empresa se centra en Madrid, según se argumentó, los clientes potenciales estarían alrededor de 10.000 – 15.000.

Seguidamente se recogen en tres tablas la **evolución prevista de las ventas**, mensualmente los dos primeros años, y trimestralmente el tercer año. Se ha calculado también el margen bruto que se obtiene.

Durante el primer año, en el que se lanza el producto, es muy poco conocido, y resulta crucial dar a conocer el producto y captar clientes rápidamente porque los costes fijos son bastante elevados para una empresa pequeña, que tendrá un capital social de 80.000€, y que presumiblemente necesitará algo de financiación.

A tal fin, los meses 4 a 7 se opta por hacer un uso intensivo de LinkedIn Ads (3.000€/mes) y de GoogleAds (otros 3.000€/mes).

La principal ventaja de LinkedIn Ads es que permite llegar a audiencias profesionales específicas pertenecientes a los segmentos de mercado objetivo, que muy probablemente al menos harán uso de la prueba gratuita del primer mes y dado el bajo precio finalmente se decidirán por mi software. Se configurarán campañas dirigidas a personas en roles como Logistics Manager o Operations Planner y empresas en sectores como transporte, distribución, servicios de entrega, gestión de incidencias o mantenimiento.

Respecto a Google Ads, utilizando campañas basadas en palabras clave relevantes como "software optimización de rutas" o "gestión logística eficiente," se atraen usuarios que ya están buscando soluciones como la mía. A diferencia del SEO (que tarda meses en mostrar resultados), Google Ads genera tráfico desde el primer momento, ideal para captar clientes rápidamente tras el lanzamiento.

No obstante, el cuarto mes se recurrirá a una auditoría externa SEO por 1.500€, con la idea de trabajar en SEO en paralelo y reducir la dependencia de anuncios en el futuro. En sectores como el de software, las palabras clave pueden tener alta competencia orgánica, lo que hace que el SEO sea un esfuerzo a medio-largo plazo. Además, SEO necesita meses de optimización (contenido, backlinks, etc.) para posicionarse en los primeros resultados.

Basándonos en la ventaja de precio, una oferta gratuita atractiva y estrategias publicitarias, podríamos ajustar las tasas de conversión al alza:

- **LinkedIn Ads:**

El pago por clic (CPC) promedio suele oscilar entre 4 y 10 € para mercados B2B. Considerando un CPC de 7 € y una tasa de conversión del 12%, con 3.000€ pueden generarse 428 clics, que supondrían la captación de 51 clientes.

- **Google Ads:**

El CPC promedio en el sector SaaS (Software as a Service), puede oscilar entre 1 y 4 € para palabras clave relevantes. Con un presupuesto de 3.000 € y un CPC promedio de 2 €, se pueden generar aproximadamente 1.500 clics. Con una tasa de conversión del 10% se podrían captar 150 clientes.

- **Marketplaces (GitHub y Gumroad):**

Subiendo el software a los marketplaces en categorías visibles (por ejemplo, herramientas de productividad o software de gestión logística), se puede atraer un tráfico significativo (hay que tener en cuenta que son plataformas internacionales).

Si consideramos que podrían atraerse un promedio de 2.000 visitas al mes combinadas (un número razonable para productos nuevos con buenas descripciones y etiquetas relevantes), con una tasa de clics (CTR) del 10% y una tasa de conversión del 10% motivada por un mes de prueba gratuito y destacando un precio competitivo, inicialmente se podrían captar alrededor de 10 clientes en los marketplaces.

Con el tiempo, a medida que el software obtenga reseñas positivas y gane visibilidad dentro de los marketplaces (por algoritmos de ranking), el tráfico puede aumentar, así como las conversiones. En 2-3 meses, es razonable que el tráfico mensual se duplique, lo que aumentaría los clientes potenciales a 20-30 clientes/mes solo desde marketplaces.

- **Publicaciones gratuitas en redes sociales y en foros**

También se captarán, aunque en mucha menor medida, clientes a través de publicaciones gratuitas en redes sociales y foros, tratando de destacar por la calidad de los contenidos, y centrándome en grupos y foros relevantes para los segmentos objetivo.

La estrategia que se seguirá consistirá en crear publicaciones con consejos generales sobre optimización de rutas o eficiencia logística y, al final, mencionar mi software como una solución práctica.

Teniendo en cuenta las estimaciones que se han realizado, durante los doce primeros meses en los que hay facturación es razonable considerar que un total de 250 clientes nuevos

utilizarán el producto cada mes. Lógicamente, habrá clientes que dejen de pagar por el producto al no estar satisfechos con el mismo, pero se espera compensar estos clientes que se pierden con las recomendaciones de los usuarios que lo encuentren útil y valioso para sus empresas.

Los siguientes doce meses, el producto ya será más conocido, tendrá una mayor visibilidad en los marketplaces, y con algo de suerte contará con buenas valoraciones en foros especializados y en comparadores de software. Las recomendaciones de los clientes captados a lo largo del primer año facilitan también incremental el volumen de ventas. Teniendo en cuenta estas consideraciones se asumen que 325 nuevas empresas más adquirirán el producto cada mes (un 30% de empresas captadas cada mes más que el año anterior).

Finalmente llega un momento en el que ya se habrán captado gran parte de los potenciales clientes, pasándose a considerar que al mes compran el producto 150 nuevas empresas, realizando así una previsión de ventas conservadora.

Considerando el precio fijado en 10€ y teniendo en cuentas las comisiones de GitHub (5%) y Gumroad (10%), si se asume una comisión promedio del 7,5%, el margen bruto por cliente es 9,25€.

Evolución de las ventas previstas durante el año 2026												
Mes	1	2	3	4	5	6	7	8	9	10	11	12
Ventas	0	0	0	0	0	250	500	750	1.000	1.250	1.500	1.750
Margen bruto	0,00	0,00	0,00	0,00	0,00	2.312,50	4.625,00	6.937,50	9.250,00	11.562,50	13.875,00	16.187,50

Evolución de las ventas previstas durante el año 2027												
Mes	1	2	3	4	5	6	7	8	9	10	11	12
Ventas	2.250	2.500	2.750	3.000	3.325	3.650	3.975	4.300	4.625	4.950	5.275	5.600
Margen bruto	20.812,50	23.125,00	25.437,50	27.750,00	30.756,25	33.762,50	36.768,75	39.775,00	42.781,25	45.787,50	48.793,75	51.800,00

Evolución de las ventas previstas durante el año 2029												
Mes	1	2	3	4	5	6	7	8	9	10	11	12
Ventas	5.925	6.250	6.575	6.900	7.050	7.200	7.350	7.500	7.650	7.800	7.950	8.100
Margen bruto	54.806,25	57.812,50	60.818,75	63.825,00	65.212,50	66.600,00	67.987,50	69.375,00	70.762,50	72.150,00	73.537,50	74.925,00

Tabla 1.- Evolución de ventas prevista durante los tres primeros años de la empresa, 2026, 2027 y 2028. Fuente: elaboración propia.

Para tener una primera valoración de la viabilidad de la empresa con este precio y esta previsión de ventas resulta interesante determinar el punto de equilibrio (break-even point). Su determinación implica resolver la siguiente ecuación:

$$\text{costes fijos} = \text{margen bruto por cliente} \times \text{n}^{\circ} \text{ clientes punto equilibrio}$$

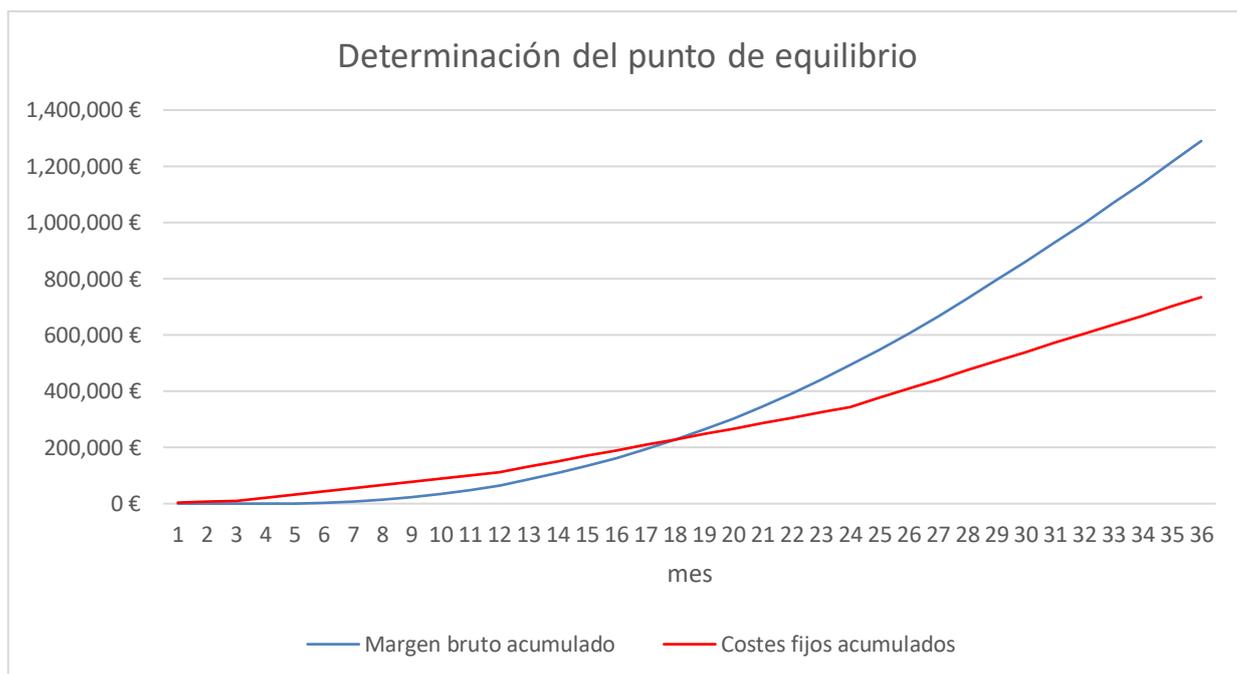


Fig. 20.- Determinación gráfica del punto de equilibrio a partir de los valores acumulados de margen bruto y costes fijos. Fuente: elaboración propia.

En el gráfico se observa que el punto de equilibrio se prevé que se alcance el mes 18 (a mediados del año 2027). A partir de entonces el margen bruto se impone frente a los costes fijos.

#### **4.4 Modelo comercial y marketing.**

En este apartado del plan de negocio se abordan todos los elementos relativos a la comunicación genérica de la propuesta de valor de la empresa, así como la captación de clientes.

##### **4.4.1 Plan de comunicación.**

La razón de ser de mi negocio es optimizar la logística de las empresas a través de una solución tecnológica sencilla y accesible, que permite generar paquetes de trabajo eficientes y sostenibles para sus empleados, maximizando el uso del transporte público y mejorando la gestión de prioridades y tiempos.

Los valores de marca que se desean transmitir son la simplicidad, costes accesibles y sostenibilidad.

El mercado objetivo de mi publicidad son empresas pequeñas y medianas con necesidades logísticas básicas, como organización de rutas de transporte para empleados, pero que buscan soluciones asequibles y fáciles de implementar sin comprometer la eficiencia. No menos interesantes, aunque sí menores en número, entre los clientes de la empresa también estarán trabajadores autónomos y freelancers, y organizaciones sin fines de lucro o comunitarias.

La marca a utilizar será “EcoRutas”, que potencia la idea de sostenibilidad, no solo por buscar rutas óptimas sino también por fomentar el transporte público. El logo que se propone se muestra en la siguiente figura.



*Fig. 21.- Logo de la marca (EcoRutas). Fuente: elaboración propia.*

Los medios de comunicación a emplear incluirían principalmente marketing digital, priorizando redes sociales y una página web que muestre contenido de valor para los segmentos de

mercado objetivo (inbound marketing), que permitirá captar clientes interesados en soluciones sostenibles.

En la página web se ofrecerá información sobre la importancia de la optimización de rutas, medio ambiente y sostenibilidad, y del software ofertado.

Dentro de la página web corporativa se integrará un knowledge base (base de conocimiento), donde se almacenarán y organizarán información, documentación y recursos relevantes, de gran interés para los clientes. Su propósito es permitir a los usuarios acceder de manera sencilla a información útil y específica, ya sea para resolver problemas, aprender a usar un producto o mejorar la eficiencia operativa.

Las redes sociales que se utilizarán serán LinkedIn, para contactar con perfiles profesionales que pertenezcan a los segmentos considerados y Facebook, haciendo visible la marca en grupos relacionados con la sostenibilidad.

También se tendrá presencia foros y comunidades relevantes relacionados con logística y optimización, muy frecuentados y útiles para promocionar una herramienta como la desarrollada. Entre estos cabe destacar Logistics World, Foro Logística España (ADL), Logistics & Delivery Forum o SCM Globe Forum.

Tal y como se mencionó en el apartado 4.3.3 Plan de desarrollo y lanzamiento de productos y servicios, se utilizará Zoom o Microsoft Teams para realizar demostraciones por videoconferencia de las funcionalidades nuevas incorporadas en las sucesivas versiones. Igualmente se hará uso de encuestas breves para valorar la experiencia de los usuarios con el producto integradas dentro del software.

A través del correo electrónico, se remitirá un newsletter con novedades, futuras mejoras y otra información que pueda resultar de interés a los clientes.

#### 4.4.2 Canales de venta.

Los canales de venta que se utilizarán son plataformas online, especialmente apropiadas por su alcance y facilidad de uso. Más concretamente, las plataformas **GitHub** y **Gumroad** son excelentes canales de venta para un software de optimización de rutas debido a su capacidad para llegar a audiencias altamente específicas y relevantes.



*Fig. 22.- Logos de Gumroad (izda.) y GitHub (dcha.), en cuyos marketplaces se venderá el software desarrollado. Fuentes: Gumroad. (n.d.). Sell what you know and see what sticks. <https://gumroad.com/> y GitHub. (n.d.). Where the world builds software. <https://github.com/>*

*GitHub Marketplace* es una plataforma ideal para llegar a desarrolladores y empresas tecnológicas. Al ser parte del ecosistema de GitHub, donde millones de programadores comparten y colaboran en proyectos de software, el código podría captar la atención de un público especializado que valora soluciones prácticas y eficientes. Además, GitHub

Marketplace permite listar el código en categorías específicas, ofreciendo visibilidad dentro de un nicho técnico, y facilita las integraciones con proyectos existentes de los usuarios. Documentando bien el código, puede atraer a usuarios que buscan soluciones listas para implementar.

Por otro lado, *Gumroad* es una plataforma popular para creadores que desean vender productos digitales, como software, libros electrónicos y cursos. Su simplicidad es una gran ventaja: permite crear una página de producto atractiva con pocos clics, procesar pagos de manera segura, y gestionar licencias o descargas automáticas para los clientes. Además, ofrece herramientas de marketing integradas, como códigos de descuento y seguimiento de ventas, lo que te permite promocionar tu código de manera efectiva. Al no requerir conocimientos técnicos avanzados para configurarla, Gumroad es ideal para lanzar rápidamente el producto al mercado.

El uso de *fuerzas de ventas externas* como agentes puede ser una estrategia complementaria, pero presenta desafíos. Aunque los agentes podrían expandir el alcance a mercados menos digitales o más tradicionales, dependen en gran medida de su capacidad de comprender el producto y transmitir su valor, lo que podría ser un reto si el software es complejo o requiere demostraciones técnicas. Además, este enfoque suele implicar mayores costes variables (comisiones) y menos control sobre el proceso de venta. En comparación, plataformas como GitHub y Gumroad no solo permiten mayor control, sino también resultados más predecibles, especialmente en las etapas iniciales del negocio, cuando la prioridad es alcanzar una base de clientes sólida en mercados tecnológicos. Por tanto, aunque las fuerzas de ventas externas pueden ser útiles para ciertos nichos, GitHub y Gumroad son una opción más estratégica para maximizar la efectividad en los primeros meses del lanzamiento.

Se baraja la posibilidad de ofrecer una comisión a agentes externos del 20-30% por venta realizada, en función del volumen de ventas que puedan generar, e incentivos adicionales si alcanzan ciertas metas de ventas. No obstante, únicamente se considerará recurrir a agentes externos en caso de que los clientes captados a través de estas plataformas resulten muy por debajo de los previstos.

En lo que a los **costes asociados** a los canales de venta se refiere:

- *Gumroad* cobra una comisión del 10% por cada venta realizada, reducida al 8.5% para usuarios que facturen más de \$1,000 mensuales, además de una tarifa fija de \$0.30 por transacción. Es económico y eficiente para ventas directas, aunque está más orientado a consumidores generales.
- En *GitHub Marketplace*, los desarrolladores deben aceptar una comisión del 5% sobre los ingresos generados por ventas a través de la plataforma. Tiene comisiones bajas y un público interesado en herramientas que mejoren su flujo de trabajo o resuelvan problemas técnicos.

#### 4.5 Modelo de relación con clientes.

En este apartado describo como se estructuran las interacciones con los clientes, que se recogen en el “customer journey map”, que plasma en una representación visual el proceso por el que pasan los clientes cuando interactúan con la empresa.

En la **fase de descubrimiento** los clientes potenciales consultan la página web anteriormente mencionada, con contenido de valor e información del producto, conocen el producto mediante redes sociales (Linkedin y Facebook).

Durante la **fase de consideración**, se consultan distintos foros relacionados con la logística y la optimización, como por ejemplo Foro Logística España (ADL), en los que estudian opiniones de productos relacionados disponibles en el mercado. Además, se recurren a comparadores de software como Capterra, especializado en herramientas para pequeñas y medianas empresas. Adicionalmente se ofrece una prueba gratuita del producto durante un periodo de un mes, lo que permite valorar si realmente el producto cubre las necesidades de la empresa.

Tal y como se explicó en el apartado anterior, la **fase de compra** se realiza a través de las plataformas GitHub y Gumroad.

En la **fase de retención** cobra importancia un newsletter que se remite por correo electrónico trimestralmente a los usuarios del producto informando de novedades, futuras mejoras, etc., así como un knowledge base con información de utilización del producto y que incluye FAQs.

Finalmente, durante la **fase de recomendación**, el cliente utiliza los medios comunicación sociales (social media) y el boca a boca.



*Fig. 23.- Descripción de las interacciones de los clientes con la empresa sintetizada en el "customer journey map". Fuente: elaboración propia.*

El momento de la verdad es el paso de la consideración a la compra. Se estima que para productos digitales se convierten en clientes tan solo un 10% de los interesados.

Es por ello que es necesario reforzar la fase de consideración, en la que es determinante la experiencia real de la empresa con el producto durante la prueba gratuita.

Se contemplan las siguientes garantías para el producto ofertado:

- Actualizaciones del producto que incluirán mejoras y nuevas opciones gratuitas.

- Tiempo garantizado para resolución de incidencias inferior a 24h.
- Cumplimiento con estándares de privacidad y seguridad de los datos proporcionados por el cliente.

Para valorar el grado de vinculación con los clientes recurriremos al Net Promoter Score (NPS) y a la Tasa de Retención de Ingresos (Revenue Retention Rate). Estos dos indicadores, que

forman parte del Cuadro de Mando Integral (CMI) de la empresa, se explican en detalle en el apartado siguiente.

No se contempla ningún tipo de segmentación o clasificación de clientes a la hora de niveles de entrega de servicios, atención y respuesta a incidencias.

Los costes previstos asociados al modelo de relación con clientes son los derivados del personal de soporte técnico y de los desarrolladores, que solucionarán errores o problemas con el código informados en las incidencias.

#### 4.6 Modelo de operaciones.

El modelo de operaciones de EcoRutas combina la utilización de transporte público y vehículos particulares para generar soluciones de desplazamiento asequibles, simples de implementar y adaptadas a las necesidades de pequeñas y medianas empresas. Al ofrecer una herramienta que mejora la gestión de tiempos y prioridades, EcoRutas promueve la reducción de emisiones de CO<sub>2</sub> y el ahorro de costes operativos. La propuesta de valor de la empresa se refuerza con un modelo disruptivo de precios que facilita la captación de clientes en un nicho competitivo. Además, el enfoque continuo en la innovación tecnológica, como la futura incorporación de bicicletas y herramientas de cálculo ambiental, posiciona a EcoRutas como líder en sostenibilidad y movilidad urbana eficiente.

Resulta conveniente alinear los objetivos, indicadores (KPIs) y metas en torno a cuatro perspectivas clave: financiera, cliente, procesos internos y aprendizaje y crecimiento. Para ello se recurre al **Cuadro de Mando Integral (CMI)**. Más allá de evaluar únicamente los resultados económicos, el CMI integra factores cualitativos y cuantitativos, proporcionando una visión holística del desempeño empresarial. Esto facilita la toma de decisiones informadas y asegura que todas las áreas de la empresa trabajen en conjunto hacia el logro de su visión estratégica.

El CMI propuesto para la empresa EcoRutas se muestra a continuación:

Perspectiva	Objetivo estratégico	Indicador	Meta	Iniciativas clave	Periodicidad
Financiera	Expansión de la empresa	Ingresos recurrentes mensuales (MRR)	18.500€ final primer año, incremento 9% mensual segundo año y del 3% el tercer año	Campañas marketing enfocadas en el bajo precio, publicaciones en web y recomendaciones foros	mensual
	Maximizar valor de cada cliente (fidelización)	Tasa de retención de ingresos (Revenue Retention Rate)	≥ 85% dos primeros años y ≥ 95% tercer año	Soporte técnico eficiente, actualizaciones y encuestas de feedback	mensual
	Incrementar la rentabilidad operativa	Margen sobre ventas (ROS)	≥ 0,5 en 2028	Optimizar costes en nuevas versiones	mensual
Cliente	Incrementar la captación de clientes	Número de clientes activos	≥ 8.000 en 2028	Estrategias de marketing para captar PYMEs	trimestral
	Mejorar la satisfacción de clientes	Net Promoter Score (NPS)	≥ 90 puntos	Encuestas al cliente	trimestral
Procesos internos	Asegurar la calidad del software	Tasa de errores críticos en el software	≤1%	Pruebas de calidad y solución de fallos reportados por los usuarios	mensual
	Integrar nuevas funcionalidades	Número de funcionalidades nuevas	≥ 2 al año	año 2025 tiempos acumulados transporte y ahorro emisiones CO2	anual
	Presencia del software en más ciudades	Ciudades españolas con más de 1.000 clientes	3 en 2029 y 6 en 2030	Campañas marketing enfocadas en el bajo precio, publicaciones en web y recomendaciones foros	anual
Aprendizaje y crecimiento	Fomentar la capacitación y desarrollo del equipo	Horas de formación por empleado	≥ 30 horas/año	Programas de formación tecnología y sostenibilidad	anual
	Incrementar la inversión en innovación	% de ingresos invertidos en I+D	≥ 10%	Funcionalidades para medir impacto ambiental Seguimiento tendencias e intereses de los clientes	anual
	Promover una cultura de sostenibilidad	Número de iniciativas ambientales implementadas	≥ 3 anuales	Campañas internas y colaboración con instituciones	anual

Fig. 24.- CMI de EcoRutas que sintetiza los indicadores clave relacionados con los objetivos estratégicos de la empresa, las metas propuestas, iniciativas clave y periodicidades de evaluación. Fuente: elaboración propia.

Conviene aclarar la interpretación de algunos **KPIs** y cómo se van a evaluar:

Los **ingresos recurrentes mensuales (MRR)**, miden los ingresos previsible y recurrentes generados por suscripciones activas al software.

$$\text{MRR} = \text{Número de clientes activos} \times \text{Precio mensual promedio por cliente.}$$

La **Tasa de Retención de Ingresos (Revenue Retention Rate, abreviado RRR)** es un indicador clave en las empresas de modelo de negocio basado en ingresos recurrentes, como el SaaS (Software as a Service). Este indicador mide cuánto de los ingresos recurrentes de un período anterior se retiene en el siguiente, tomando en cuenta renovaciones, cancelaciones y expansiones de los clientes existentes (inexistentes en por el momento para mi producto).

Una alta tasa de retención indica que los clientes permanecen vinculados a la empresa, renovando sus contratos o suscripciones. Un RRR bajo puede ser una señal de problemas en el producto, servicio al cliente o insatisfacción general, indicando la necesidad de mejorar la experiencia del cliente.

$$\text{RRR} = (\text{Ingresos del inicio del período} - \text{Pérdidas de ingresos por cancelaciones}) \div \text{Ingresos del inicio del período.}$$

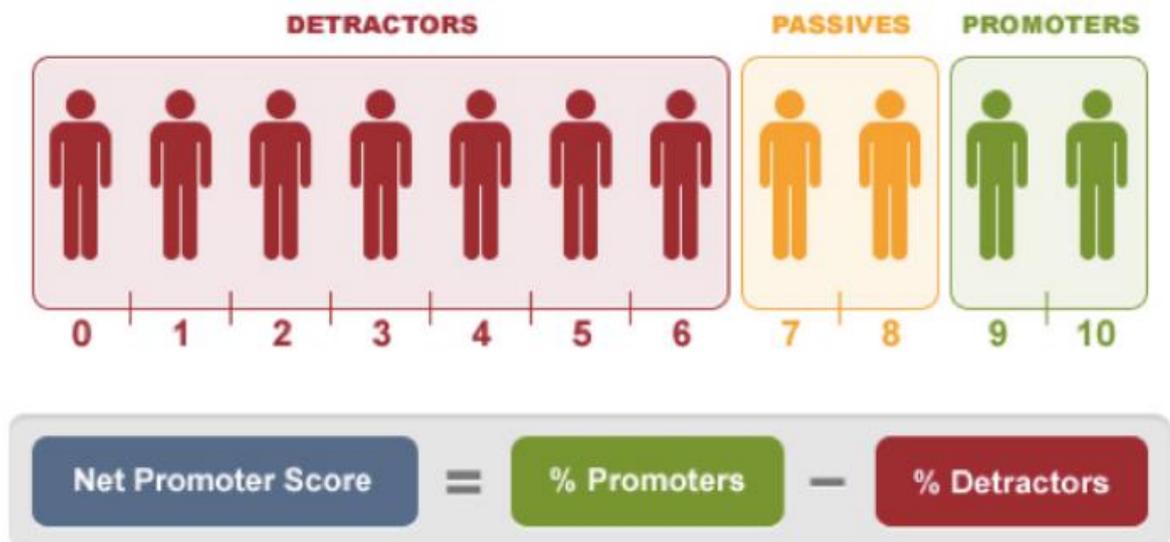
El **Net Promoter Score (NPS)** se determina a partir de los resultados de la encuesta de satisfacción de clientes.

Un promotor en NPS es cualquier usuario que haya respondido a la pregunta "¿con qué probabilidad recomendaría este producto?" con una puntuación de 9 o 10.

Un detractor en NPS es cualquier usuario que haya respondido a la pregunta "¿con qué probabilidad recomendaría este producto?" con una puntuación entre 0 y 6 (incluido).

Conocidos los promotores y detractores se evalúa el NPS como:

$$\text{NPS} = \% \text{ promotores} - \% \text{ detractores}$$



*Fig. 25.- Clientes promotores, pasivos y detractores en función de la puntuación del producto en la encuesta de satisfacción. Fuente: SignificaCloud. (n.d.). Voz del empleado, voz del cliente y NPS. <https://www.meaningcloud.com/es/blog/voz-del-empleado-voz-del-cliente-y-nps>*

La **tasa de errores críticos en el software** se expresará como el número de incidencias en las que la ejecución del código ha sido deficiente (error de ejecución o no se ajusta a especificaciones de usuario) respecto al total de ejecuciones por parte del conjunto de clientes.

$$\text{Tasa de errores (\%)} = (\text{errores críticos detectados} / n^0 \text{ veces hacen uso clientes}) \times 100$$

#### 4.7 Partners y alianzas.

En este apartado se identifican los proveedores clave para nuestro negocio indicando, así mismo, si existe alguno de ellos que suponga una dependencia significativa del modelo de negocio. Igualmente se consideran posibles partners o alianzas que pudiesen resultar de interés.

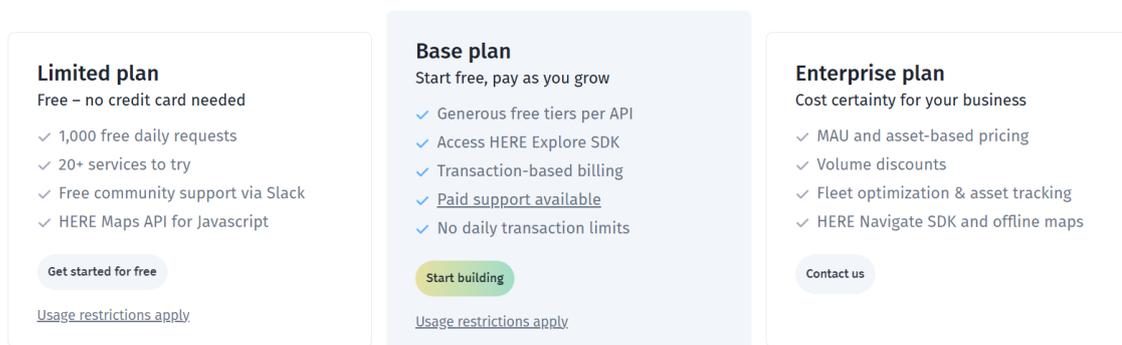
El **proveedor clave** actualmente es Google, ya que el código se ha desarrollado haciendo uso de las APIs de Google Maps. Aunque el pricing de su API directions se repercute a los clientes, cambios en su política de pricing puede llevar a que las empresas que tienen que cubrir un volumen más elevado de desplazamientos tengan que pagar a Google una cantidad adicional a los 10€/mes por la utilización del producto, que puede llegar a ser nada desdeñable. Este hecho supondría una depreciación del valor percibido del producto ofertado muy considerable y un gran daño a la imagen de la empresa.

En vistas del problema que puede suponer una dependencia excesiva con este proveedor, ya se han considerado otras APIs **alternativas** como **Openrouteservice**, para la cual ya se conoce cómo debería modificarse el código (ver apartado 3.4.2) y cuyas principales ventajas son límites diarios gratuitos generosos (más económico que Google Maps) y ser de código abierto, lo que permitiría la personalización y adaptación específica a las necesidades del producto.

Otra opción que resulta interesante para consultar distancias y tiempos en transporte público es **Public Transit Routing API**, de HERE Technologies (forma parte de su suite Public Transit

API v8). Tiene un modelo de precios competitivo, especialmente en grandes volúmenes de uso y su cobertura en Europa es muy amplia, aunque su integración es más compleja. Esta API podría combinarse con HERE Routing API, que permite calcular rutas optimizadas para ciclistas, teniendo en cuenta factores como carriles bici, rutas ciclistas seguras y la geografía del terreno.

Los planes ofertados por HERE Technologies se muestran a continuación:



Plan	Key Features	Call to Action
<b>Limited plan</b> Free – no credit card needed	<ul style="list-style-type: none"><li>✓ 1,000 free daily requests</li><li>✓ 20+ services to try</li><li>✓ Free community support via Slack</li><li>✓ HERE Maps API for Javascript</li></ul>	Get started for free
<b>Base plan</b> Start free, pay as you grow	<ul style="list-style-type: none"><li>✓ Generous free tiers per API</li><li>✓ Access HERE Explore SDK</li><li>✓ Transaction-based billing</li><li>✓ <a href="#">Paid support available</a></li><li>✓ No daily transaction limits</li></ul>	Start building
<b>Enterprise plan</b> Cost certainty for your business	<ul style="list-style-type: none"><li>✓ MAU and asset-based pricing</li><li>✓ Volume discounts</li><li>✓ Fleet optimization &amp; asset tracking</li><li>✓ HERE Navigate SDK and offline maps</li></ul>	Contact us

Fig. 26.- Planes de HERE Technologies. Fuente: HERE Technologies. (n.d.). Pricing plans for location services. <https://www.here.com/get-started/pricing>

Podrían combinarse las APIs de HERE Technologies para transporte público y bicicleta y la API Directions de Google Maps para desplazamientos en coche. Esto permitiría a los clientes manejar un mayor número de emplazamientos sin incurrir en costes adicionales.

Para desplazamientos en coche se puede recurrir también a la API de TomTom, muy potente en cálculo de rutas y tiempos de viaje, bien documentada y con modelos de precios competitivos.

Finalmente se abordan posibles **alianzas estratégicas**:

- *Empresas de bicicletas compartidas, transporte público o scooters eléctricos*: podría integrar sus servicios directamente en mi herramienta, lo que no solo enriquecería la experiencia del cliente, sino que también me permitiría acceder a su base de usuarios y una promoción conjunta.
- *Instituciones que promuevan el transporte sostenible*: podrían ayudarme a posicionar mi producto como una solución de interés social.
- *CRMs (Customer Relationship Management)*: la integración de mi herramienta con CRMs ya populares como Salesforce, HubSpot, etc. podría ampliar el alcance del producto ofertado, ya que clientes actuales de estas plataformas podrían acceder fácilmente a mi solución.
- *Comunidades verdes*: colaborar con ellas mejorará la diferenciación de mi producto y lo hará más visible para potenciales clientes comprometidos con el medio ambiente.

#### 4.8 Gestión de talento.

En este apartado se incluye una descripción de la organización de la empresa y todos los elementos relativos a lo que tradicionalmente se conoce como gestión de recursos humanos. En concreto: una breve descripción de sus posiciones clave, qué actividades generales planteamos externalizar a través de contratos de servicio, el plan de contratación de personas y los sistemas de remuneración, compensación, beneficios y reconocimiento del desempeño.

El **organigrama de la empresa** cuenta con tres departamentos, programación, tecnologías de la información (TI) y marketing, que suman un total de 6 empleados (yo formo parte del Dpto. de programación). Puede observarse en la siguiente figura, en la que se sintetizan las funciones de los distintos departamentos.

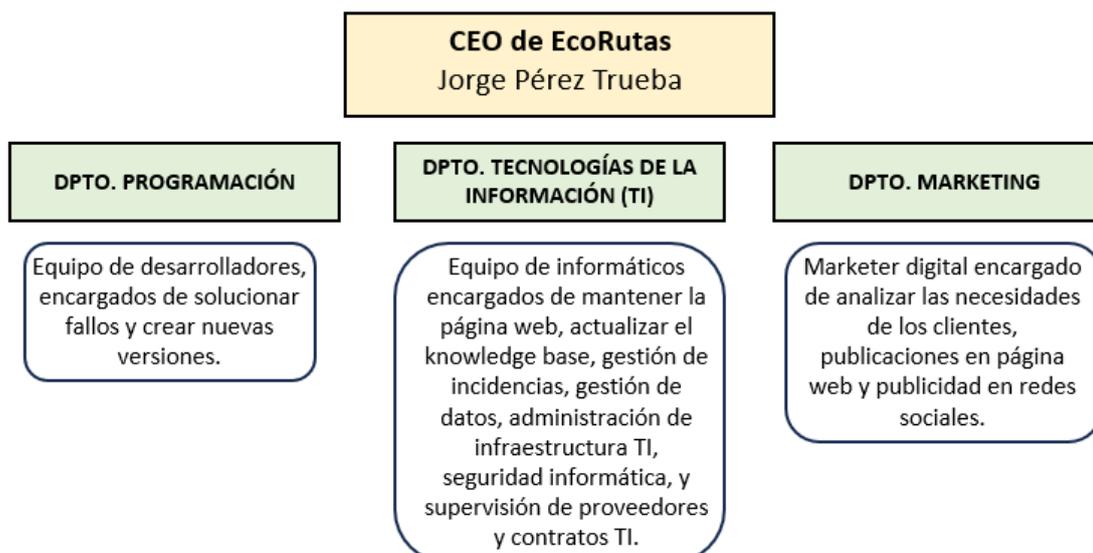


Fig. 27.- Organigrama propuesto para la empresa. Fuente: elaboración propia.

En cuanto al **talento clave** a atraer a la compañía, se especifica seguidamente para los distintos departamentos:

- Departamento de programación: desarrollador Backend Python con conocimientos avanzados en algoritmos de optimización y manejo de librerías como Pandas, Googlemaps y Datetime, así como habilidad para integrar APIs externas relacionadas con mapas (al menos Google Maps).
- Departamento de tecnologías de la información (TI):
  - Dominio para modificar, optimizar o solucionar problemas de la interfaz web (HTML, CSS o JavaScript) y conocimientos en frameworks de frontend
  - Conocimientos sólidos en SQL y gestión de bases de datos.
  - Experiencia en administración de servidores y despliegue en la nube (Google Cloud).
  - Habilidades en configuración y mantenimiento de infraestructuras TI.
  - Experiencia en diseño e implementación de políticas de seguridad informática.
  - Imprescindible tener conocimientos en Python
- Departamento de marketing: especialistas en marketing digital con conocimientos amplios en posicionamiento SEO/SEM, habilidad en redes sociales (especialmente LinkedIn y Facebook), dominio de herramientas como Google Ads y Google Analytics. Además, se valorarían muy positivamente conocimientos de sostenibilidad y marketing verde.

Intentaré que la empresa sea **atractiva para las nuevas incorporaciones** destacando el impacto positivo en el medio ambiente de un proyecto innovador que ofrece oportunidades de crecimiento en la empresa debido a su escalabilidad. Otro aspecto muy valorado que ofrece la empresa es la posibilidad de trabajar en remoto, lo que favorece el equilibrio entre la vida personal y profesional.

Además, se fomentará un ambiente donde todas las ideas sean valoradas, haya comunicación abierta entre los distintos departamentos y todos se sientan partícipes del éxito de la empresa (se ofrecerán igualmente bonos por desempeño de hasta el 15% del salario bruto).

Por último, caben destacar los **beneficios sociales** que contemple la empresa:

- Abono a los empleados de un máximo de 20 €/mes por realización de viajes a través de Bicimad, que muy seguramente será suficiente para cubrir la totalidad de los desplazamientos realizados en bicicleta por la ciudad de Madrid.
- Abono de transporte para sus trabajadores hasta la zona en que se encuentren sus domicilios. En caso de vivir fuera de la ciudad de Madrid se dará una ayuda de 30 €/mes.
- Seguro de salud a los empleados. El costo mensual por empleado podría situarse aproximadamente entre 30€ y 60€.

La empresa potenciará cuatro **valores** muy importantes, que resultan clave para su éxito:

- **Sostenibilidad:** la empresa tiene muy presente el medio ambiente. Potenciar este valor no solo conecta con las tendencias actuales del mercado, sino que también atrae a clientes y empleados comprometidos con su preservación. La sostenibilidad debe ser una bandera que guíe todas las decisiones de negocio, desde el diseño del producto hasta las operaciones internas.
- **Facilidad de uso y enfoque en el cliente:** la facilidad de uso debe estar en el centro de todas las acciones de la empresa. Es necesario escuchar a los clientes, entender sus necesidades, y anticiparse a sus expectativas para mejorar el producto, y construir una relación sólida y de confianza con ellos.
- **Innovación:** es uno de los pilares fundamentales para mantener la competitividad en el mercado y garantizar que el producto ofrecido evolucione con las necesidades de los clientes. No basta solo con trabajar en la mejora continua del producto, sino que también se debe fomentar una mentalidad creativa y de experimentación en todo el equipo.
- **Colaboración y equipo:** dado que los departamentos de programación, TI y marketing deben trabajar juntos, es crucial fomentar la colaboración. Un equipo cohesionado puede resolver problemas de forma más efectiva, innovar con rapidez y generar un ambiente laboral saludable.

Es necesario **apoyar la cultura y valores de la empresa** estableciendo una visión clara y compartida a todos los empleados a través de reuniones periódicas.

Como CEO de la empresa, seré el primero en actuar conforme a sus valores, implementando prácticas como el fomento del transporte sostenible o el trabajo remoto para reducir emisiones, tal y como quedó reflejado al tratar los beneficios sociales.

Los valores deben de incorporarse también desde el Proceso de Selección y Onboarding. Durante la contratación, se buscarán personas que no solo tengan las habilidades técnicas necesarias, sino que también compartan los valores de la empresa. En el proceso de incorporación, dedica tiempo a explicar la misión, visión y cultura organizacional, para que los nuevos empleados se alineen desde un principio.

Para fomentar la innovación interna se recurrirá a dinámicas para que los empleados aporten ideas y experimenten con nuevas soluciones, como, por ejemplo, hackathons internos (encuentro de programadores en el que se busca, a través de su trabajo colaborativo, dar una respuesta a un reto o problema), talleres creativos o sesiones de lluvia de ideas. Además, se recompensarán las iniciativas que contribuyan a mejorar el producto o las operaciones mediante los bonos por desempeño.

El trabajo en equipo se promoverá a través de jornadas de voluntariado ambiental como reforestación, limpieza de ríos o parques locales. También se organizarán excursiones en la naturaleza agregando elementos de juego, como búsqueda del tesoro, resolución de acertijos o tareas colaborativas como construir un refugio para pernoctar.

Respecto al **esquema retributivo** que se propone, teniendo en cuenta que se buscarán perfiles con al menos 3 años de experiencia, es el siguiente:

- Especialistas en marketing digital: salario bruto anual de 30.000 €/año.
- Desarrolladores backend Python: salario bruto anual de 48.000 €/año.
- Ingenieros informáticos del departamento IT: salario bruto anual de 39.000 €/año.

No obstante, el primer año, para reducir los costes salariales, y dado que yo trabajaré como desarrollador, optaré por ajustar mi salario bruto a 25.000 €/año.

Lógicamente, el coste de trabajadores es algo superior porque hay que tener en cuenta conceptos relacionados con la cuota de la Seguridad Social que no aparecen reflejados en el salario bruto como son contingencias comunes (cubren la futura pensión del trabajador y sus correspondientes tipos de baja laboral) y profesionales (cubren las incapacidades temporales o permanentes derivadas de accidentes o enfermedades asociadas de forma directa al puesto de trabajo), prestaciones por desempleo, el Fondo de Garantía Salarial (FOGASA) por si la empresa se declarase insolvente y la formación. Se estiman así los siguientes costes:

Costes trabajadores de la empresa	
Desarrollador Backend Python	5.400 €/mes.empleado
Especialista mkt digital	3.400 €/mes.empleado
Ingeniero informático IT	4.400 €/mes.empleado

Tabla 2.- Costes previstos de los distintos perfiles de los trabajadores de la empresa. Fuente: Impulsa Empresa. (n.d.). Coste del trabajador para la empresa <https://www.impulsa-empresa.es/coste-trabajador-empresa/>

Costes previstos de los distintos perfiles de los trabajadores de la empresa. Fuente: elaboración propia.

El coste empresa asociado a mi trabajo como desarrollador, se reduce a 2.800 €/mes durante el primer año.

Aunque durante los tres primeros años la empresa mantendrá el mismo organigrama, los **costes de personal** irán evolucionando debido a la sucesiva incorporación de trabajadores como consecuencia de la necesidad de apoyo para desarrollar las nuevas versiones y de atender a un volumen de clientes creciente.

El primer año trabajaré únicamente yo como desarrollador, con un sueldo más bajo para reducir costes, junto con un ingeniero informático IT y un marketer digital. El segundo año se incorporará a la empresa un segundo desarrollador y me subiré el sueldo bruto a 48.000 €/año, que es el que se ha establecido para el puesto de desarrollador. Ante el considerable aumento que se espera que experimente la empresa en el número de clientes a lo largo del segundo año, el tercer año se contratará a otros tres ingenieros informáticos IT.

Los **costes operativos** pueden aumentar de forma progresiva, aunque no de forma muy significativa, debido a las necesidades de alojamiento en la nube, capacidad de la base de datos o de un plan más completo de la herramienta para gestionar facturación y contabilidad. En función de decisiones del departamento de marketing, con vistas a captar más clientes,

también se podría incurrir en un mayor gasto por campañas realizadas. Por simplicidad, y teniendo en cuenta que no se espera que experimenten un aumento considerable como para

alterar la viabilidad ni el retorno del negocio, se asumirán constantes e iguales a los determinados para el primer año.

En cuanto a las decisiones de **subcontratación y externalización**, ya se ha comentado en el apartado en el apartado 4.3.3, que el desarrollo de la interfaz de usuario, necesaria para poder lanzar el producto se subcontratará y supondrá incurrir en un gasto de 3.000€ – 7.000€.

Respecto a la externalización, es muy probable para la gestión de facturación y contabilidad en función de la evolución del volumen de clientes.

La decisión de subcontratar una empresa para desarrollar la interfaz de usuario viene motivada, en el ahorro de tiempo (se desea lanzar el producto en cuatro meses), el acceso a expertos altamente cualificados y evitar posibles errores que el equipo de desarrolladores podría cometer debido a falta de experiencia, como problemas de rendimiento, diseño inconsistente o no adaptativo. Además, la subcontratación permite que los desarrolladores puedan centrarse en áreas donde tienen más experiencia, como el backend y los algoritmos de optimización, así como ajustar los recursos a corto plazo, sin necesidad de contratar personal adicional.

Durante los tres primeros años, al no tener demasiados clientes y con el fin de evitar incurrir en un gasto excesivo, tal y como se comentó con anterioridad, se recurrirá al uso de una herramienta para gestionar facturación y contabilidad, encargándome yo mismo de ello. A partir del tercer año, la empresa puede llegar a tener más de 10.000 clientes, por lo que **externalizar** la gestión de facturación y contabilidad puede ser una decisión estratégica para la empresa. Se barajarán empresas como Adecco Outsourcing S.L. o Avantges Outsourcing S.L., que ofrecen estos servicios.

#### 4.9 Sistemas de información.

En este apartado se incluye la identificación de la infraestructura de informática y comunicaciones necesaria para soportar las operaciones, los procesos de negocio, la gestión económico-financiera, los procesos de recursos humanos, servicios de puesto de trabajo, todas las herramientas asociadas al perfil digital de la empresa y los procesos de continuidad de negocio.

La **infraestructura de informática personal (hardware y software)** que se contempla es:

- **Hardware:** portátiles de alto rendimiento con procesadores i9, 32 GB de RAM, SSD de 512 GB, pantallas Full HD y accesorios (ratones ergonómicos y estaciones docking).
- **Software:** sistemas operativo Windows 11 Pro, Microsoft 365, herramientas de desarrollo Visual Studio Code, y navegadores seguros (Chrome y Edge).
- **Seguridad:** antimalware Bitdefender, VPN corporativa, y cifrado de disco duro.

La empresa opta por el **servidor** en la nube de Google Cloud (más capas de seguridad respecto a un servidor local tradicional). El **almacenamiento** también se realizará en la nube mediante Google Cloud, realizándose igualmente **copias de seguridad** en local.

Respecto a la **infraestructura de comunicaciones**, dado que los empleados de la empresa pueden trabajar en remoto, es decisión suya su red (LAN/Wi-Fi) y sus características, aunque se recomienda conexión a internet de alta velocidad con redundancia (mínimo 500 Mbps simétricos). Para las **telecomunicaciones** se optará por las plataformas Microsoft Teams y Zoom.

Para un startup, como esta empresa, no se considera necesario un **ERP**, aunque si resultaría necesario, tal y como se comentó con anterioridad, alguna herramienta enfocada en la contabilidad y finanzas, que facilite la **gestión administrativo – financiera**, que incluya funciones como facturación, seguimiento de gastos, impuestos, nóminas y reportes financieros (por ejemplo, QuickBooks).

Aunque las plataformas como Gumroad y GitHub ya ofrecen funcionalidades básicas para gestionar clientes y ventas, que podrían ser suficiente en una primera instancia, un **CRM** puede ser muy ventajoso para seguimiento de leads, mantener un historial de interacciones con clientes, automatizar campañas de marketing para promover las nuevas versiones y analizar datos (segmentar la base de clientes y estudiar métricas clave, como la tasa de conversión). La opción que se baraja es HubSpot CRM, ya que su versión gratuita, que incluye muchas características útiles como integraciones con correos electrónicos, seguimiento de interacciones y reportes, podría ser suficientes para mi empresa.

Los elementos que darán **soporte al perfil digital de la compañía** serán la web corporativa, como núcleo de la presencia digital de la empresa, complementado con un knowledge base, esencial para ofrecer soporte a clientes y usuario, y un newsletter para mantener una comunicación continua con los clientes.

El posicionamiento web se mejorará mediante estrategias SEO y campañas publicitarias en Google Ads para aumentar el tráfico hacia el sitio web. La presencia en redes sociales también será un pilar importante, especialmente LinkedIn y Facebook. Por otra parte, también contribuye a la presencia digital los marketplaces de GitHub y Gumroad, a través de los cuales se oferta el producto.

Para garantizar la **continuidad de negocio** y la **recuperación frente a desastres** se contempla definir y probar un Disaster Recovery Plan, que podría incluir replicación en tiempo real contratando AWS CloudEndure (costo estimado de 20€/mes por servidor), monitoreo de la red, servidor y base de datos para detectar fallos (Zabbix es una buena opción porque es gratuito y en la empresa habrá personal con conocimientos técnicos para configurarlo), y copias de seguridad automáticas con objetivos RPO/RTO bien definidos.

Sería conveniente **externalizar** la gestión de servidores y almacenamiento en la nube, incluso teniendo un departamento de TI, ya que permite a mi equipo interno centrarse en tareas estratégicas y de mayor valor, como la actualización del knowledge base, la gestión de incidencias y de datos. Analizando el mercado, se llega a la conclusión de que empresas como Jotelulu o CenterServ pueden cobrar entre 100€-300€/mes dependiendo de la complejidad.

Finalmente se resumen los gastos operativos asociados a IT y comunicaciones, así como en el gasto previsto para equipos informáticos y su plan de amortización.

Gastos operativos IT y comunicaciones	
Servidor	100€/mes
VPN (5 usuarios)	10€/mes por usuario
Gestión servidor y almacenamiento en la nube	200€/mes
Servicio replicación en tiempo real	20€/mes
Plan básico QuickBooks (herramienta de facturación y contabilidad)	30€/mes
Alojamiento en la nube	100€/mes
Base de datos SQL	75€/mes
Dominio web profesional	5€/mes
Cuenta de correo profesional	8€/mes cada cuenta
Windows 11 pro	16€/mes cada licencia
Microsoft 365	20€/mes cada licencia

Tabla 3.-Estimación de los gastos operativos en IT y comunicaciones de la empresa EcoRutas. Fuente: elaboración propia.

Inversión en equipos informáticos	
Portátil	2.250€/unidad
Ratón ergonómico	40€/unidad
Estación docking	45€/unidad

Tabla 4.- Estimación de la inversión necesaria en equipos informáticos de los trabajadores. Fuente: elaboración propia.

Los portátiles de alto rendimiento, suelen volverse obsoletos en un plazo de 3 a 5 años debido a los avances tecnológicos, nuevas exigencias de software, y desgaste físico.

La tecnología asociada (ratones y estaciones docking) también tiene una vida útil similar o ligeramente inferior.

Por otra parte, según establece el art. 12.1.a de la Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades, para los equipos electrónicos el coeficiente lineal máximo es el 20%, mientras que el periodo de años máximo para realizar amortización es 10 años.

TABLA DE COEFICIENTES DE AMORTIZACIÓN LINEAL		
Equipos electrónicos e informáticos. Sistemas y programas.	Coeficiente lineal máximo	Periodo de años máximo
Equipos electrónicos.	20%	10
Equipos para procesos de información.	25%	8
Sistemas y programas informáticos.	33%	6
Producciones cinematográficas, fonográficas, videos y series audiovisuales	33%	6
Otros elementos	10%	20

Tabla 5.- coeficientes de amortización lineal establecidos por la Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades. Fuente: España. (2014). Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades. Boletín Oficial del Estado (BOE), núm. 288, de 28 de noviembre de 2014, pp. 96939-97058. <https://www.boe.es/buscar/doc.php?id=BOE-A-2014-12328>

Se decide realizar una amortización lineal en 5 años, que implica una depreciación anual de 467€/año para cada equipo.

## **4.10 Modelo económico.**

Este capítulo refleja de forma clara la evolución del flujo de caja generado por las operaciones, la evolución de la rentabilidad de la compañía, a través de la cuenta de pérdidas y ganancias, y la evolución de su equilibrio patrimonial, a través del balance de situación.

### **4.10.1 Flujos de caja.**

En este apartado se reflejar la evolución del flujo de caja neto de la compañía desde su constitución como cobros menos pagos, la evolución del nivel de caja, para determinar las necesidades máximas de tesorería a cubrir con financiación, y el momento temporal en que se empieza a generar flujo de caja positivo.

Los cobros y pagos de los tres primeros años se detallan a continuación, mientras que las tablas con la evolución mensual de la caja se adjuntan en el Anexo III.

#### **Primer año (2026):**

La empresa se constituirá con un capital social de 80.000€. Yo seré el socio principal, aportando un capital de 30.000€. El capital restante se prevé conseguir captando otros 10 socios que aporten 5.000€ cada uno. Además, para enfrentar la fase de lanzamiento y ante el escaso margen bruto de los primeros meses, que afectarán significativamente a la caja se pedirá un préstamo de 40.000€

Para poder lanzar el producto al mercado, se necesita subcontratar a una empresa que se encargue de desarrollar la interfaz de usuario. Consideraré que el gasto asociado será de 4.500€.

El producto se lanza al mercado el mes 4, en el que todavía no se tiene ningún ingreso ya que se ofrece una versión gratuita de un mes para darlo a conocer y hacerlo más atractivo.

Durante los meses 4 a 7, es crítica la promoción del producto, por lo que se recurre a Google Ads (previéndose un gasto de 3.000€/mes) y a LinkedInAds (otros 3.000€/mes).

El cuarto mes se recurrirá a una auditoría externa SEO por 1.500€, con vistas a dejar de pagar por el servicio Google Ads.

Los trabajadores de la empresa este primer año, a partir del mes 4 son: yo mismo como desarrollador backend Python, un especialista en marketing digital y un ingeniero informático IT. El coste de la masa salarial asciende a 10.600€/mes. Los tres primeros meses, únicamente trabajaré yo como desarrollador (2.800€).

La inversión en los tres equipos informáticos de los trabajadores es de 7.005€.

Los gastos operativos de IT y comunicaciones que se prevén ascienden a 712€/mes.

Al final del ejercicio no se reparten dividendos a los accionistas dado que este ejercicio la empresa incurre en unas pérdidas considerables.

### **Segundo año (2027):**

Cerrado el año fiscal, el mes de enero, se pagan 11.512,5€ de bonificaciones (supongo que se han cumplido los objetivos).

Para facilitar el desarrollo de las nuevas versiones y que se puedan lanzar en los plazos previstos se requiere contratar un desarrollador adicional, que supone un gasto adicional a la empresa de 5.400€/mes. Es necesario comprar por tanto otro equipo informático de 2.335€.

Este segundo año me igualo el sueldo al del desarrollador adicional contratado, 48.000€ de sueldo bruto. El gasto de la empresa por los dos desarrolladores pasa ser de 10.800€/mes.

Los gastos operativos de IT y comunicaciones se mantienen este segundo año, en total 712€/mes.

Cerrado el ejercicio se reparten 40.000€ de dividendos de forma directamente proporcional a las aportaciones de los socios.

### **Tercer año (2028):**

Cerrado el año fiscal, el mes de enero, se pagan 24.750€ de bonificaciones (supongo que se han cumplido los objetivos).

Dado que el volumen de clientes ha crecido de forma considerable se decide contratar en enero a otros tres ingenieros informáticos IT. El coste empresa de los trabajadores pasa a ser de 31.800 €/mes.

Además, se adquieren tres equipos informáticos adicionales, que implican una inversión de 7.005€.

Cerrado el ejercicio se reparten 160.000€ de dividendos de forma directamente proporcional a las aportaciones de los socios.

A principios del cuarto año debe pagarse en concepto de bonos de desempeño 30.600 €, considerando que cumplan con los objetivos que se les proponga

#### **4.10.2 Balance de situación.**

Realizados los asientos contables (ver anexo IV. Asientos contables de los tres primeros años) en las distintas cuentas de activo, pasivo y patrimonio neto, los balances de situación de los tres primeros ejercicios son los siguientes:

BALANCE A 31 DE DICIEMBRE DE 2026	
<b>ACTIVO</b>	
Caja	39.298,47
Activo corriente	39.298,47
Activo inmovilizado	5.954,25
Activo no corriente	5.954,25
<b>Total activo</b>	<b>45.252,72</b>
<b>PASIVO + PATRIMONIO NETO</b>	
Impuestos	0,00
Bonos por pagar	11.512,50
Pasivo corriente	11.512,50
Deuda a largo	27.456,24
Pasivo no corriente	27.456,24
Socios	80.000,00
Reservas	-73.716,02
Patrimonio neto	6.283,98
<b>Total pasivo + patrimonio neto</b>	<b>45.252,72</b>
FM	27.785,97
NOF	-11.512,50

Tabla 6.- Balance de EcoRutas a 31 de diciembre de 2026. Fuente: elaboración propia.

En 2025, el balance presenta un activo total de 45.252,72€, compuesto mayoritariamente por el activo corriente (39.298,47€, correspondiente al saldo de caja). El activo no corriente, con 5.954,25€, representa una proporción pequeña del total. En cuanto al pasivo y patrimonio neto, la empresa tiene una deuda a corto plazo 11.512,50€ en bonos de desempeño por pagar y una deuda a largo plazo de 27.456,24€. El patrimonio neto, con apenas 6.283,98€, refleja como unas reservas negativas de -73.716,02€ reducen significativamente el capital propio.

El fondo de maniobra (FM) es 27.785,97€, lo que indica que la empresa tiene un exceso de activo corriente para cubrir sus pasivos corrientes. Es decir, con el fondo de maniobra disponible la empresa puede satisfacer completamente sus necesidades de capital operativo sin requerir fuentes externas de financiación.

En este año, aunque el patrimonio neto es bajo (6.283,98€), el balance refleja una posición financiera razonablemente estable. Sin embargo, la alta proporción de deuda a largo plazo (27.456,24€) puede limitar la flexibilidad financiera a futuro.

BALANCE A 31 DE DICIEMBRE DE 2027	
<b>ACTIVO</b>	
Caja	166.454,44
Activo corriente	166.454,44
Activo inmovilizado	6.421,25
Activo no corriente	6.421,25
<b>Total activo</b>	<b>172.875,69</b>
<b>PASIVO + PATRIMONIO NETO</b>	
Impuestos	14.098,03
Bonos por pagar	24.750,00
Pasivo corriente	38.848,03
Deuda a largo	14.138,81
Pasivo no corriente	14.138,81
Socios	80.000,00
Reservas	39.888,84
Patrimonio neto	119.888,84
<b>Total pasivo + patrimonio neto</b>	<b>172.875,69</b>
FM	127.606,40
NOF	-38.848,03

Tabla 7.- Balance de EcoRutas a 31 de diciembre de 2027. Fuente: elaboración propia.

En 2027, el activo total crece significativamente hasta 172.875,69€, impulsado principalmente por el aumento del activo corriente, que alcanza 166.454,44€ (casi cuatro veces más que en 2025). El activo no corriente se mantiene en 6.421,25€. En el lado del pasivo, se observa un incremento importante en las deudas a corto plazo (38.848,03€), donde aparecen impuestos (14.098,03€) y bonos por pagar aumentan a 24.750,00€. Sin embargo, la deuda a largo plazo disminuye a 14.138,81€.

El crecimiento del activo corriente (166.454,44€) está acompañado por un aumento en el pasivo corriente (38.848,03€), pero la relación sigue siendo favorable gracias al elevado FM. Además, la disminución de la deuda a largo plazo (14.138,81€) refleja una gestión prudente del endeudamiento.

El patrimonio neto mejora significativamente (119.888,84€) gracias al incremento de las reservas (39.888,84€), lo que indica que la empresa está acumulando recursos propios y mejorando su solidez financiera.

El aumento significativo del FM a 127.606,40€, y las NOF = -38.848,03€ confirman que la empresa sigue operando sin requerimientos adicionales de financiación operativa. Esto es un indicador positivo, ya que el exceso de liquidez generado puede ser utilizado para reinversión o reducción de deuda.

BALANCE A 31 DE DICIEMBRE DE 2028			
<b>ACTIVO</b>		<b>PASIVO + PATRIMONIO NETO</b>	
Caja	353.667,37	Impuestos	54.245,37
Activo corriente	353.667,37	Bonos por pagar	42.300,00
Activo inmovilizado	10.157,25	Pasivo corriente	96.545,37
Activo no corriente	10.157,25	Deuda a largo	0,00
<b>Total activo</b>	<b>363.824,62</b>	Pasivo no corriente	0,00
		Socios	80.000,00
		Reservas	187.279,26
		Patrimonio neto	267.279,26
		<b>Total pasivo + patrimonio neto</b>	<b>363.824,62</b>
FM	257.122,01		
NOF	-96.545,37		

Tabla 8.- Balance de EcoRutas a 31 de diciembre de 2028. Fuente: elaboración propia.

Para 2028, el activo total alcanza 363.824,62€, más del doble que en 2027. Este crecimiento se explica por el incremento del activo corriente, que asciende a 353.667,37€, y del activo no corriente por la compra de equipos informáticos de los ingenieros IT que se incorporan a la empresa (10.157,25€). En cuanto al pasivo, los impuestos por pagar aumentan considerablemente (52.245,37€), y los bonos por pagar también crecen a 42.300,00€, lo que eleva el pasivo corriente a 96.545,37€. Sin embargo, desaparecen las deudas a largo plazo.

El patrimonio neto mejora aún más, situándose en 267.279,26€, gracias al incremento sustancial de las reservas (187.279,26€).

En este ejercicio, el fondo de maniobra alcanza 257.122,01€, cubriendo ampliamente el pasivo corriente y reflejando una posición de liquidez muy sólida. Las NOF = -96.545,37 indican que la empresa tiene un buen manejo de su financiación operativa, ya que no depende de liquidez propia o externa para sostener sus operaciones a corto plazo.

Además, la ausencia de deuda a largo plazo a partir de este año elimina cualquier presión financiera futura relacionada con obligaciones de largo plazo, fortaleciendo aún más la posición financiera.

Finalmente se comenta brevemente la **evolución de los balances** y se justifica un FM tan elevado en 2028.

La evolución de los balances muestra un crecimiento constante en el activo total, impulsado por aumentos en el efectivo. El fondo de maniobra cubre perfectamente las necesidades operativas, eliminando la necesidad de financiamiento adicional para sostener las operaciones.

El patrimonio neto mejora sustancialmente cada año, gracias a la acumulación de reservas, lo que refuerza la solidez financiera de la empresa y permitirá acometer inversiones y el crecimiento de la empresa sin recurrir a financiación externa.

Aunque las deudas a corto plazo aumentan progresivamente, estas están completamente cubiertas por el activo corriente, lo que no genera riesgos de liquidez. Sin embargo, sería prudente controlar su crecimiento para evitar que supere la capacidad de cobertura en ejercicios futuros.

El crecimiento del fondo de maniobra en cada ejercicio refleja una gestión eficiente del capital de trabajo. En 2028, la eliminación de la deuda a largo plazo y la acumulación de reservas consolida una posición financiera muy fuerte, proporcionando una gran flexibilidad para futuras inversiones o proyectos de expansión.

Se aprovechará el fondo de maniobra (FM) elevado para financiar la expansión de la empresa, que hasta el momento se ha centrado en darse a conocer en Madrid, provincia en la que se prevé que ya se hayan captado una gran parte de los potenciales clientes. Los años 2029 y 2030 EcoRutas se centrará en la expansión en otros grandes centros urbanos de España, Barcelona y Valencia en 2029; y Sevilla, Málaga y Bilbao en 2030.

La empresa tiene recursos suficientes para cubrir los costos iniciales de la expansión, como la adaptación del software e inversión en marketing y publicidad, sin necesidad de recurrir a financiamiento externo. Esto reduce los riesgos financieros.

Si bien expandirse siempre implica riesgos, EcoRutas tiene un colchón financiero que le permite afrontar posibles contratiempos sin comprometer sus operaciones.

Con una ejecución adecuada, esta expansión podría ser una gran oportunidad para posicionar la empresa como uno de los líderes en su sector y aumentar significativamente sus ingresos en el mediano y largo plazo.

#### 4.10.3 Cuenta de resultados.

En este apartado se presenta la cuenta de resultados al final de cada año de los tres primeros años de operación.

RESULTADOS AÑO 2026	
Ventas	90.000,00
CMV	-6.750,00
Margen bruto	83.250,00
Gastos operativos (Opex)	-153.856,50
Beneficio operativos (Ebitda)	-70.606,50
Resultado de explotación (Ebit)	-71.657,25
Beneficio antes de impuestos	-73.716,02
Beneficio neto	-73.716,02

Tabla 9.- Cuenta de resultados de EcoRutas del año 2026. Fuente: elaboración propia.

RESULTADOS AÑO 2027	
Ventas	462.000,00
CMV	-34.650,00
Margen bruto	427.350,00
Gastos operativos (Opex)	-258.362,00
Beneficio operativos (Ebitda)	168.988,00
Resultado de explotación (Ebit)	167.120,00
Beneficio antes de impuestos	165.834,90
Beneficio neto	151.736,87
ROS	0,33
ROE	24,15

Tabla 10.- Cuenta de resultados de EcoRutas del año 2027. Fuente: elaboración propia.

RESULTADOS AÑO 2028	
Ventas	862.500,00
CMV	-64.687,50
Margen bruto	797.812,50
Gastos operativos (Opex)	-435.713,00
Beneficio operativos (Ebitda)	362.099,50
Resultado de explotación (Ebit)	358.830,50
Beneficio antes de impuestos	358.366,79
Beneficio neto	304.121,42
ROS	0,35
ROE	2,54

Tabla 11.- Cuenta de resultados de EcoRutas del año 2028. Fuente: elaboración propia.

La evolución de los resultados de la empresa entre los años 2026, 2027 y 2028 muestra un crecimiento significativo en los ingresos y una mejora sustancial en los márgenes y la rentabilidad.

En 2026, la empresa experimentó una situación financiera desfavorable, con unas ventas de 90.000€ y un beneficio neto negativo de -73.716,02€. Este resultado refleja una estructura de gastos operativos elevada (153.856,50€) y la lógica dificultad inicial para captar clientes, lo que provocó un EBITDA negativo (-70.606,50€) y un EBIT también negativo (-71.657,25€). Este año muestra una falta de equilibrio entre los ingresos y los costos operativos.

Sin embargo, en 2027, la empresa registra una notable recuperación ya que va consiguiendo captar clientes progresivamente. Las ventas se multiplican por más de cinco, alcanzando los 462.000€. Este aumento de los ingresos permite generar un margen bruto de 427.350€, aunque los gastos operativos también aumentan hasta los 258.362€, principalmente debido a mi subida de sueldo y la contratación de otro desarrollador.

A pesar de este incremento en gastos, el EBITDA y el EBIT son positivos, alcanzando 168.988€ y 167.120€, respectivamente. Finalmente, el beneficio neto se sitúa en 148.419,65€. Además, se calculan indicadores clave de rentabilidad como el ROS (0,33) y el ROE (24,15), que reflejan una gestión más eficiente de los recursos.

En 2028, el crecimiento continúa de manera exponencial. Las ventas aumentan a 862.500€, lo que supone un incremento significativo respecto al año anterior. El margen bruto se sitúa en 797.812,50€, y los gastos operativos, aunque también crecen hasta 435.713,00€, no afectan negativamente a la rentabilidad. El EBITDA duplica su valor respecto a 2026, alcanzando 362.099,50€, mientras que el EBIT es de 358.830,50€. El beneficio neto llega a 304.121,42€, consolidando una posición de rentabilidad sólida. El ROS mejora ligeramente a 0,35, y aunque el ROE disminuye a 2,54, esto es debido al considerable aumento del patrimonio neto (119.888,84€) de la empresa en 2027 por el incremento de las reservas (39.888,84€).

En resumen, la evolución de los resultados refleja un crecimiento sostenido en las ventas y una mejora en la gestión de los costos, lo que lleva a la empresa de una situación crítica en 2026 a una rentabilidad destacable en 2027 y 2028.

#### **4.11 Resumen ejecutivo**

EcoRutas es una empresa innovadora que ofrece un software de optimización de rutas de fácil utilización, que permite generar paquetes de trabajo eficientes y sostenibles para los empleados de las empresas, maximizando el uso del transporte público frente a los vehículos de combustión y mejorando la gestión de prioridades y tiempos.

El mercado objetivo son pequeñas y medianas empresas con necesidades logísticas básicas, que desarrollan su actividad en entornos urbanos, que buscan soluciones asequibles y fáciles de implementar sin comprometer la eficiencia, y que además consideran que el transporte público puede ser una alternativa interesante debido a los crecientes problemas de tráfico en las grandes poblaciones y a las limitaciones de acceso a determinadas zonas de los vehículos más contaminantes.

La creación de esta empresa supone una disrupción en un nicho específico del mercado que requiere la optimización de rutas para transporte con una estructura de necesidades heterogéneas, integrando transporte público y automóviles particulares, en el que se compite además en precio. Los planes del principal competidor, Route4Me, comienzan en 36€/mes, mientras que el precio del producto desarrollado por la empresa será de tan solo 10€/mes, lo que presumiblemente ayudará a captar clientes rápidamente. Puede afirmarse por tanto que el producto que se ofrece tiene el potencial de crear un océano azul.

Cabe destacar además que la mejora continua del software a través de sucesivas versiones que incorporarán nuevas funcionalidades como la incorporación de la bicicleta como medio de transporte alternativo y el cálculo del ahorro de emisiones de CO<sub>2</sub> posicionará el producto como uno de los softwares líderes en sostenibilidad.

Respecto a los resultados financieros esperados, basados en una previsión de ventas conservadora (solo en la provincia de Madrid hay unas 10.000 empresas potenciales y el tercer año se han considerado menos de 9.000 clientes, un número bajo para un precio tan competitivo), cabe destacar lo siguiente:

- **Crecimiento Exponencial en Ingresos:** las ventas estimadas aumentarán de 90.000€ en 2026 a 462.000€ en 2027, y alcanzarán 862.500€ en 2028.
- **Mejora Sostenida en la Rentabilidad:** la empresa pasará de un beneficio neto negativo en 2026 (-73.716,02€) a beneficios netos crecientes de 151.736,87€ en 2027 y 304.121,42€ en 2028, destacando su efectividad en la gestión de costes y optimización de operaciones.
- **Indicadores Financieros Sólidos:** el margen sobre ventas (ROS), próximo a 0,35 los años 2027 y 2028, y el retorno sobre patrimonio (ROE), superior a 2,5 estos dos años, evidencia una gestión eficiente de los recursos.

En cuanto a la solidez de los balances y posición financiera de la empresa, en 2028 EcoRutas logrará eliminar completamente su deuda a largo plazo, dejando un patrimonio neto de 267.279,26€ y contará con un fondo de maniobra sólido de 257.122,01€, que no solo cubrirá ampliamente el pasivo corriente, sino que también proporcionará la liquidez necesaria para su expansión.

EcoRutas planea aprovechar su fortaleza financiera para expandirse a otras grandes ciudades en España, incluyendo Barcelona y Valencia en 2028, y Sevilla, Málaga y Bilbao en 2029. Con un modelo probado y recursos internos disponibles para cubrir los costos iniciales de expansión (marketing, software y logística), esta estrategia promete incrementar significativamente los ingresos al abrirse a nuevas regiones con alta demanda y fortalecer la posición competitiva como líder en el sector, consolidando la marca EcoRutas como un referente en movilidad sostenible.

En resumen, EcoRutas representa una oportunidad única de inversión en una empresa que combina rentabilidad, crecimiento y sostenibilidad, consolidándose como líder emergente en su sector. Sin embargo, para poder lanzar su producto necesitan 10 socios que aporten 5.000€ cada uno para complementar los 30.000€ de capital social que aportará Jorge Pérez Trueba como fundador de la empresa.

Con una base financiera sólida y un modelo de negocio probado, EcoRutas está en condiciones de maximizar el valor para sus accionistas, que recibirán 2.500€ de dividendos al

final del segundo año y 10.000€ al final del tercero, lo que supone un VAN de 4.579,26€ considerando una tasa de retorno del 10%, y una TIR del 39%, sin tener en cuenta el valor que podría tener la empresa en 2028, estimado en 900.000€ (tres veces el beneficio neto previsto para el último ejercicio).

## Capítulo 5. Conclusión y futuras líneas de trabajo.

### 5.1 Conclusiones.

En España, alrededor de 200.000 empresas dedicadas a la logística, almacenamiento, transporte, servicios técnicos y mantenimiento enfrentan un desafío común: optimizar las rutas diarias para llevar a cabo tareas en diferentes ubicaciones.

De otra parte, los entornos urbanos enfrentan problemas de tráfico y contaminación, donde la circulación de vehículos contaminantes es predominante. El interés por alternativas sostenibles como el transporte público o la bicicleta está creciendo como consecuencia de estos problemas, y son soluciones cada vez más frecuentes por las que apuestan muchos particulares y empresas.

En un contexto donde la eficiencia operativa es esencial para la competitividad, pequeñas y medianas empresas, autónomos y freelancers suelen carecer de herramientas accesibles y asequibles para gestionar sus recursos. Además, muchos de ellos no cuentan con conocimientos avanzados en informática, lo que dificulta el uso de soluciones técnicas complejas.

Por ello, el desarrollo de un software de optimización de rutas centrado en la facilidad de uso, un precio competitivo y un enfoque en la sostenibilidad representa una solución ideal para este segmento de mercado. Su comercialización tiene un alto potencial de rentabilidad, respondiendo a una necesidad clara y creciente.

El software desarrollado contribuye significativamente a la sostenibilidad desde tres perspectivas clave: ambiental, social y económica. Desde el punto de vista ambiental, su capacidad para integrar el transporte público y la bicicleta reduce las emisiones de gases de efecto invernadero y el consumo energético asociado al transporte tradicional. Esto disminuye la huella de carbono de las empresas y de sus empleados, apoyando la transición hacia un modelo de movilidad más limpio y respetuoso con el medio ambiente.

En el ámbito social, la solución impulsa una movilidad más accesible y equitativa al facilitar herramientas asequibles para pequeñas y medianas empresas y trabajadores independientes, que a menudo carecen de los recursos para adoptar sistemas avanzados. Al optimizar rutas y fomentar el uso de transporte sostenible, se mejora la calidad de vida en entornos urbanos al reducir la congestión vehicular y el ruido, creando ciudades más habitables. Por otro lado, en el plano económico, el software fomenta una mayor eficiencia operativa al disminuir costos asociados al transporte y mejorar la productividad de las empresas. Este enfoque integral no solo refuerza la competitividad empresarial, sino que también sienta las bases para un crecimiento sostenible y responsable.

La propuesta de negocio presentada en este trabajo fin de máster consiste en la comercialización del software desarrollado. Este software integra el uso de transporte público y automóviles particulares, atendiendo necesidades específicas que las soluciones actuales no han logrado satisfacer. Aunque no busca redefinir el mercado global, combina elementos de innovación incremental y disruptiva, mejorando funcionalidades existentes y aportando un enfoque único y diferenciado.

Enfocado en un nicho poco atendido, como la optimización de rutas para empleados que utilizan medios de transporte mixtos, este producto tiene el potencial de generar un océano azul. Su propuesta de valor distintiva puede atraer a un segmento específico de clientes, diferenciándose de los competidores tradicionales y posicionándose como líder en este mercado especializado.

Mediante una diferenciación en precio respecto del principal competidor, Route4Me, se logra que pequeñas y medianas empresas que no pueden pagar tarifas tan altas se inclinen hacia el software aquí desarrollado.

Recurriendo a LinkedIn Ads, Google Ads, la mejora del posicionamiento web mediante estrategias SEO, la visibilidad que dan al software los marketplaces GitHub y Gumroad, así

como a publicaciones gratuitas en redes sociales y en foros, se prevé lograr una evolución de las ventas que permite que se alcance el punto de equilibrio en 18 meses (a mediados de 2027).

A partir de los costes previstos y de los flujos de caja esperados, se observa que, con un capital social de 80.000€, para tener caja suficiente es necesario recurrir a financiación externa, por lo que se necesitará pedir un préstamo de 40.000€.

Al analizar los balances de los tres primeros ejercicios, se observa que, aunque en 2026 la empresa enfrentará una situación financiera compleja debido a elevados gastos operativos y desafíos iniciales para captar clientes, se espera un crecimiento constante en las ventas y una optimización progresiva de los costos. Esto permitirá alcanzar una rentabilidad destacable en 2027 y 2028.

Las proyecciones financieras son muy positivas. Los ingresos crecerán significativamente, pasando de 90.000 € en 2026 a 462.000 € en 2027, y alcanzarán 862.500 € en 2028. Asimismo, la rentabilidad mejorará de forma constante, evolucionando de un beneficio neto negativo en 2026 (-73.716,02 €) a ganancias netas de 151.736,87 € en 2027 y 304.121,42 € en 2028, reflejando una gestión eficiente de los costos y una mejora operativa.

Además, la prevista expansión de EcoRutas a otras grandes ciudades españolas entre 2028 y 2029, respaldada por un modelo probado y recursos internos sólidos, permitirá aumentar considerablemente los ingresos al ingresar a regiones de alta demanda. Esto fortalecerá la posición competitiva de la empresa, consolidándola como líder en el sector y referente en sostenibilidad.

Finalmente, la empresa presenta un atractivo notable para los inversores. Una inversión de 5.000 € en el capital social generaría dividendos de 2.500 € al finalizar el segundo año y 10.000 € al término del tercero. Esto implica un VAN de 4.579,26 € considerando una tasa de retorno del 10 % y una TIR del 39 %, sin contar el valor proyectado de la empresa en 2028, estimado en 900.000 €.

## 5.2 Futuras líneas de trabajo.

El primer gran reto es captar inversores que hagan posible la empresa. Se necesitarán 10 socios que aporten 5.000€ cada uno para complementar mi aportación de 30.000€ al capital social.

El perfil de los inversores que pueden estar interesados en un software de optimización de rutas es amplio, aunque me centraré en profesionales del sector logístico y transporte, inversionistas ángeles con capacidad de inversión en busca de negocios escalables y con potencias tecnológico, y en amigos a los que les ha resultado interesante el software y la posibilidad de comercializarlo.

Para localizar a los posibles inversores se recurrirá a LinkedIn, identificando perfiles de inversores ángeles, consultores logísticos o emprendedores del sector tecnológico. También se recurrirá a incubadoras y aceleradoras de startups, y se contempla asistir a eventos de networking y ferias del sector tecnológico.

Una vez localizados, se preparará una presentación personalizada para cada inversor, ajustando la propuesta según sus intereses y prioridades. Sería conveniente asegurarse de que firmen una carta de intención o un preacuerdo una vez que estén interesados, mantenerse en contacto y responder a sus dudas rápidamente.

Tal y como se ha mencionado en el trabajo, convertir un código de Python para optimización de rutas en un software funcional y utilizable requiere más que solo la lógica algorítmica. Deben considerarse aspectos como la interfaz, el backend, la integración con otros sistemas y la experiencia del usuario. Esto constituye el primer desafío al que hay que enfrentarse para alcanzar el producto mínimo viable.

El software deseado será accesible desde la web y estará basado en la nube. Toda la lógica del sistema, el almacenamiento de datos y el procesamiento se gestionan en servidores remotos alojados en la nube de Google Cloud. Esto permite a los usuarios acceder al software con su ordenador desde cualquier lugar con conexión a internet, utilizando distintos dispositivos.

En cuanto al backend, aunque falta poco por hacer, habría que realizar modificaciones del código para poder manejar múltiples solicitudes simultáneamente.

Para la interfaz de usuario web (frontend) hay que seleccionar a una empresa externa como Velneo, APPYWEB o Zaask. Los requisitos indispensables de la interfaz es que sea intuitiva y fácil de usar que permita la carga de datos.

Falta también implementar la base de datos SQL que permitirá gestionar tanto los datos operativos como históricos del sistema.

El éxito de la empresa estará muy fuertemente condicionado por el nivel de competencia de los empleados que se contraten.

El desarrollador se buscará en plataformas como GitHub, Stack Overflow, y Reddit, donde a menudo, sus perfiles incluyen enlaces para contacto. Observar cómo escriben el código (claridad, estructura, comentarios, etc.) da una buena idea de su estilo de trabajo y nivel de detalle; por otra parte, la profundidad y precisión en las respuestas sobre temas técnicos puede ser una señal de experiencia.

Para el resto de empleados se recurrirá a portales de empleo tradicionales como InfoJobs, LinkedIn, Indeed o Glassdoor. Se publicarán ofertas detalladas que incluirán la descripción del puesto, las habilidades requeridas, y los beneficios de trabajar en mi empresa. Estos portales tienen un amplio alcance y me permitirán filtrar candidatos por experiencia y habilidades.

Finalmente, habrá que abordar el desarrollo de las próximas versiones se refiere, que harán el software más competitivo y le sumarán valor.

La versión 2 introducirá la estimación de tiempos invertidos en transporte y acumulados en cada emplazamiento según el tiempo medio introducido por el usuario. Requerirá implementar funciones para calcular tiempos acumulados como parte del backend y utilizar visualizaciones simples para mostrar resultados.

La versión 3 incorporará una estimación del ahorro de emisiones de CO<sub>2</sub> por optimización de rutas, tanto en coche como en transporte público. La estrategia que se seguirá consiste en la obtención de datos confiables sobre emisiones de CO<sub>2</sub> por kilómetro para cada tipo de transporte utilizando APIs como Carbon Interface API o bases de datos oficiales.

La versión 4 contemplará la bicicleta como transporte alternativo. Su implementación se basará en gran parte en las estrategias de las versiones 2 y 3. Muy probablemente se recurrirá a HERE Routing API, que permite calcular rutas optimizadas en bicicleta.

## ANEXOS

### Anexo I. Código de Python desarrollado.

Se importan las librerías:

```
import time
import pandas as pd
import requests
import openrouteservice
import json
import googlemaps
import datetime as dt
from datetime import datetime
```

Configuraciones iniciales:

```
# clave api de opencagedata (obtener coordenadas de una dirección)
api_key_opencagedata = '2c8b0b7e8efa43dc88a442398a165ddf'

# nombre del excel de partida
excel_direcciones = 'direcciones_reparto.xlsx'

# clave api distacematrix de Gogle Maps
api_key_googlemaps = 'AIzaSyAMH10_VN9JjGr5W_CyBPX5ufSrh1N1LDw'

Maps = googlemaps.Client(key= 'AIzaSyAMH10_VN9JjGr5W_CyBPX5ufSrh1N1LDw')
```

Se abre la Excel con los emplazamientos y se recogen en DataFrame:

```
# leer el excel y pasarlo a dataframe
df = pd.read_excel(excel_direcciones, sheet_name=0)
#Creo una copia, porque luego se borrarán las direcciones que entren en reparto
df_inicial=df.copy()

# para imprimir el data frame: print(df.head())

# Extraer la columna de direcciones como una lista
if 'DIRECCION' in df.columns:
    direcciones = df['DIRECCION'].dropna().tolist()
else:
    print('La columna dirección no se encuentra en el excel')

# Extraer la columna con la fecha de alta como una lista
if 'FECHA ALTA' in df.columns:
```

```
fecha_alta = df['FECHA ALTA'].dropna().tolist()
else:
    print('La columna fecha alta no se encuentra en el excel')
```

Se recorren las direcciones y se calculan las distancias y tiempos en transporte público respecto de la empresa:

```
direccion_empresa = str(input('¿Cuál es la dirección de la empresa (punto de partida trabajadores)? Ej:C/Bustamante 16, Madrid, Spain').strip())
current_date = dt.datetime.today()
```

#Crea una lista para almacenar las distancias respecto de la empresa

```
datos_trayecto_transporte_publico = []
datos_dias_diferencia = []
```

```
for fecha in fecha_alta:
```

```
    fecha_str = fecha.strftime("%d/%m/%Y")
    date_object = dt.datetime.strptime(fecha_str, "%d/%m/%Y")
    diferencia_fechas = abs(current_date - date_object)
    dias=diferencia_fechas.days
```

```
    datos_dias_diferencia.append(dias)
```

```
for direccion_destino in direcciones:
```

```
    # Cálculo de la distancia y tiempo recorrido andando/coche/transporte público entre la oficina y la dirección a la que es necesario asistir
```

```
    # Los modos son 'walking' 'driving' 'transit'
```

```
    distancia_transporte_publico =
```

```
Maps.directions(direccion_empresa,direccion_destino + ', Madrid, Spain',mode='transit')
```

```
    km_distancia_tp =
```

```
(distancia_transporte_publico[0]['legs'][0]['distance']['value'])/1000
```

```
    min_tiempo_tp =
```

```
(distancia_transporte_publico[0]['legs'][0]['duration']['value'])/60
```

```
    km_distancia_tp=round(km_distancia_tp,2)
```

```
    min_tiempo_tp=round(min_tiempo_tp,2)
```

```
    datos_trayecto_transporte_publico.append((km_distancia_tp,min_tiempo_tp))
```

Se crea DataFrame y Excel con las distancias y tiempos en transporte público desde la empresa:

```
#OBTENGO UN DATAFRAME, QUE LUEGO GUARDO COMO EXCEL, CON LA INFORMACIÓN DE DISTANCIA Y TIEMPO EN TRANSPORTE PÚBLICO DESDE LA EMPRESA
```

```
#ESTE EXCEL ME VA A SERVIR PARA DETERMINAR POSTERIORMENTE A QUÉ EMPLAZAMIENTOS SE IRÁ EN TRANSPORTE PÚBLICO Y A CUÁLES EN COCHE
```

```
if len(df) == len(datos_trayecto_transporte_publico): #Compruebo que las
filas del excel de partida coinciden con el tamaño de la lista con las
distancias calculadas
    #Añado al dataframe df la columna DIFERENCIA_DÍAS
    df['DIFERENCIA_DIAS']=datos_dias_diferencia
    #Desempaquetado la lista distancias en lista_andando y lista_coche
    lista_distancia_transporte_publico,lista_tiempo_transporte_publico =
zip(*datos_trayecto_transporte_publico)
    #Añado al dataframe df las columnas DISTANCIA_TRANSP_PUBLICICO y
TIEMPO_TRANSP_PUBLICICO
    df['DISTANCIA_TRANSP_PUBLICICO']=lista_distancia_transporte_publico
    df['TIEMPO_TRANSP_PUBLICICO']=lista_tiempo_transporte_publico

    #Genero un nuevo Excel con las dos nuevas columnas
    df.to_excel('Reparto_con_datos_transporte_publico.xlsx',index=False)
else:
    print('El número de líneas del Excel de partida y de la lista de
distancias son diferentes')
```

Se crean las hojas con los datos de distancias y tiempos para las direcciones en transporte público y en coche respectivamente:

```
#CREO EL DATAFRAME QUE CONTIENE LA INFORMACIÓN DE LOS EMPLAZAMIENTOS A LOS QUE
SE VA A IR EN TRANSPORTE PÚBLICO)
andando_df=df[df['TIEMPO_TRANSP_PUBLICICO']<25] #Recoge toda la fila cuando los
valores que hay en la celda de la columna TIEMPO_TRANSP_PUBLICICO son menores
que 20 min
andando_df=andando_df.sort_values(by=['PRIORIDAD','DIFERENCIA_DIAS','DISTANCIA
_TRANSP_PUBLICICO'],ascending=[True,False,True]) #Ordeno por prioridad y
dentro de esta por días transcurridos desde alta y distancia (mayor a menor y
menor a mayor respectivamente)
emp_realizar_andando = len(andando_df)

#CREO EL DATAFRAME QUE CONTIENE LA INFORMACIÓN DE LAS DIRECCIONES A LAS QUE IR
EN COCHE
coche_df=df[df['TIEMPO_TRANSP_PUBLICICO']>=25] #Recoge toda la fila cuando los
valores que hay en la celda de la columna TIEMPO_TRANSP_PUBLICICO son mayores o
iguales que 20 min
#Elimino del dataframe coche_df las columnas DISTANCIA_TRANSP_PUBLICICO y
TIEMPO_TRANSP_PUBLICICO
coche_df=coche_df.drop(columns=['DISTANCIA_TRANSP_PUBLICICO'])
coche_df=coche_df.drop(columns=['TIEMPO_TRANSP_PUBLICICO'])
#Ahora calculo los tiempos en coche a las direcciones a las que se va a acudir
en coche desde la oficina
datos_trayecto_coche = []
direcciones_coche = coche_df['DIRECCION'].dropna().tolist()
```

```
for direccion_destino in direcciones_coche:

    distancia_coche = Maps.directions(direccion_empresa,direccion_destino + ',
Madrid, Spain',mode='driving')
    km_distancia_coche =
(distancia_coche[0]['legs'][0]['distance']['value']/1000
    min_tiempo_coche = (distancia_coche[0]['legs'][0]['duration']['value']/60
    km_distancia_coche=round(km_distancia_coche,2)
    min_tiempo_coche=round(min_tiempo_coche,2)
    datos_trayecto_coche.append((km_distancia_coche,min_tiempo_coche))
#Añado las columnas de DISTANCIA_COCHE y TIEMPO_TRANSP_COCHE
if len(coche_df) == len(datos_trayecto_coche):
    lista_distancia_coche,lista_tiempo_coche = zip(*datos_trayecto_coche)
    #Añado al dataframe df las columnas DISTANCIA_TRANSP_PUBLICO y
TIEMPO_TRANSP_PUBLICO
    coche_df['DISTANCIA_COCHE']=lista_distancia_coche
    coche_df['TIEMPO_COCHE']=lista_tiempo_coche
else:
    print('El número de líneas del Excel de partida y de la lista de
distancias son diferentes')

coche_df=coche_df.sort_values(by=['PRIORIDAD', 'DIFERENCIA_DIAS', 'DISTANCIA_COC
HE'],ascending=[True,False,False])
#print(coche_df.head())
emp_realizar_coche = len(coche_df)

#CREO UN EXCEL DE DISTANCIAS CON DOS HOJAS, UNA CON LAS DIRECCIONES A
PROGRAMAR ANDANDO Y OTRA CON LAS QUE SE DEBEN PROGRAMAR EN COCHE
with pd.ExcelWriter('Distancias.xlsx') as writer:      #Flujo encargado de
lectura y escritura
    andando_df.to_excel(writer,sheet_name='Andando',index=False)
    coche_df.to_excel(writer,sheet_name='Coche',index=False)

Se calcula el reparto de direcciones a las que se irá en transporte público:

#Creo el dataframe andando_df_inicial, copia antes del original, antes de
hacer reparto
andando_df_inicial=andando_df.copy()

#Creo el dataframe andando_df_reparto que contendrá los paquetes de
direcciones a los que asistirán en transporte público los trabajadores en su
jornada laboral
andando_df_reparto=pd.DataFrame(columns=andando_df.columns)
andando_df_reparto['PESO_TIEMPO']=None
andando_df_reparto['PESO_PRIORIDAD']=None
andando_df_reparto['PESO_DIAS_TRANSCURRIDOS']=None
andando_df_reparto['PESO_TOTAL']=None
```

```
respuesta_andando = str(input('¿Desea programar emplazamientos a los que ir
andando? (SI/NO)').strip())
while respuesta_andando!="SI" and respuesta_andando!="NO":
    respuesta_andando = str(input('Escriba una respuesta
válida (SI/NO)').strip())

while respuesta_andando=="SI":

    num_paquetes_andando = input('¿Cuántos paquetes de direcciones andando
quieres?').strip()
    num_paquetes_andando = int(num_paquetes_andando)

    emp_andando_trabajador = input('¿A cuántas direcciones andando va a ir
cada trabajador en su jornada?').strip() #strip hace que no se recojan
espacios blancos
    emp_andando_trabajador = int(emp_andando_trabajador)

    for i in range(num_paquetes_andando):

        direccion_origen=andando_df['DIRECCION'].iloc[0] #Toma el valor de
la columna 'DIRECCION' fila 0 (primera)
        clave_primaria_origen=andando_df['CLAVE PRIMARIA'].iloc[0]

        distancias_desde_primer_punto = []
        tiempos_desde_primer_punto = []

        for j in range(1,len(andando_df)):

            direccion_destino=andando_df['DIRECCION'].iloc[j]

            distancia_transporte_publico =
Maps.directions(direccion_origen,direccion_destino + ', Madrid,
Spain',mode='transit')
            km_distancia_tp =
(distancia_transporte_publico[0]['legs'][0]['distance']['value'])/1000
            min_tiempo_tp =
(distancia_transporte_publico[0]['legs'][0]['duration']['value'])/60
            km_distancia_tp=round(km_distancia_tp,2)
            min_tiempo_tp=round(min_tiempo_tp,2)

            distancias_desde_primer_punto.append(km_distancia_tp)
            tiempos_desde_primer_punto.append(min_tiempo_tp)

        #Creo un dataframe que contendrá los pesos desde la dirección origen
        andando_df_pesos = andando_df.drop('DISTANCIA_TRANSP_PUBLICICO',axis=1)
        andando_df_pesos =
andando_df_pesos.drop('TIEMPO_TRANSP_PUBLICICO',axis=1)
```

```
    andando_df_pesos=andando_df_pesos.iloc[1:]

    andando_df_pesos['DISTANCIA_TRANSP_PUBLICO']=distancias_desde_primer_punto
unto
    andando_df_pesos['TIEMPO_TRANSP_PUBLICO']=tiempos_desde_primer_punto

    #Determino los pesos y los vuelco a dataframe andando_df_pesos
    pesos_tiempos=[]
    pesos_prioridades=[]
    pesos_dias=[]
    pesos_totales=[]

    min_tiempo=min(tiempos_desde_primer_punto)
    max_tiempo=max(tiempos_desde_primer_punto)

    dias_transcurridos=andando_df_pesos['DIFERENCIA_DIAS'].dropna().tolist
()
    min_dias=min(dias_transcurridos)
    max_dias=max(dias_transcurridos)

    for z in range(len(distancias_desde_primer_punto)):

        peso_tiempo = round(10 - (9/(max_tiempo -
min_tiempo))*(tiempos_desde_primer_punto[z]-min_tiempo),2) # El primer peso
saldrá mayor a 10, pero no va a afectar el resultado
        pesos_tiempos.append(peso_tiempo)

        peso_prioridad = round(10 - andando_df['PRIORIDAD'].iloc[z],2)
        pesos_prioridades.append(peso_prioridad)

        peso_dia = round(10 - (9/(max_dias -
min_dias))*(dias_transcurridos[z]-min_dias),2)
        pesos_dias.append(peso_dia)

        peso_total = peso_tiempo + peso_prioridad + peso_dia
        pesos_totales.append(peso_total)

    andando_df_pesos['PESO_TIEMPO']=pesos_tiempos
    andando_df_pesos['PESO_PRIORIDAD']=pesos_prioridades
    andando_df_pesos['PESO_DIAS_TRANSCURRIDOS']=pesos_dias
    andando_df_pesos['PESO_TOTAL']=pesos_totales

    andando_df_pesos=andando_df_pesos.sort_values(by=['PESO_TOTAL'],ascending=[False])

    for index in range(emp_andando_trabajador):
```

```
if(index==0): #LOS TIEMPOS Y DISTANCIAS DE LA PRIMERA FILA SON
DESDE OFICINA!!

    fila=andando_df.iloc[[index]] #Para recoger la fila
como un dataframe
    fila.loc[:, 'PESO_TIEMPO' ] = [10]
    fila.loc[:, 'PESO_PRIORIDAD' ] = [10]
    fila.loc[:, 'PESO_DIAS_TRANSCURRIDOS' ] = [10]
    fila.loc[:, 'PESO_TOTAL' ]= [30]

    andando_df_reparto=pd.concat([andando_df_reparto, fila], ignore_
index=True)

    andando_df=andando_df[andando_df['CLAVE
PRIMARIA']!=clave_primaria_origen] #Se queda con todo el dataframe menos la
que tiene el expediente origen
    andando_df=andando_df.reset_index(drop=True) #Para
reenumerar los índices del dataframe (queda un índice sin contenido tras
borrar)

    df=df[df['CLAVE PRIMARIA']!=clave_primaria_origen]
    df=df.reset_index(drop=True)

else: #LOS TIEMPOS Y DISTANCIAS RESTANTES SON DESDE LA PRIMERA
DIRECCIÓN A LA QUE VA EL TRABAJADOR!!

    fila=andando_df_pesos.iloc[[index-1]]

    andando_df_reparto=pd.concat([andando_df_reparto, fila], ignore_
index=True)

    clave_primaria=fila['CLAVE PRIMARIA'].values[0] #.values[0]
porque fila es un dataframe con una única fila, por tanto el número de expte
está en el índice cero

    andando_df=andando_df[andando_df['CLAVE
PRIMARIA']!=clave_primaria]
    andando_df=andando_df.reset_index(drop=True)

    df=df[df['CLAVE PRIMARIA']!=clave_primaria_origen]
    df=df.reset_index(drop=True)

    andando_df_reparto.loc[len(andando_df_reparto)] = [None] *
len(andando_df_reparto.columns)

    respuesta_andando = input('¿Desea programar nuevos paquetes de trabajo
andando? (SI/NO)').strip()
```

```
while respuesta_andando!="SI" and respuesta_andando!="NO":
    respuesta_andando = str(input('Escriba una respuesta
válida (SI/NO)').strip())
```

Se calcula el reparto de direcciones a las que se irá en coche:

```
#Creo el dataframe coche_df_inicial, copia antes del original, antes de hacer
reparto
coche_df_inicial=coche_df.copy()
```

```
#Creo el dataframe coche_df_reparto que contendrá los paquetes de de
direcciones a los que asistirán en coche los trabajadores en su jornada
laboral
```

```
coche_df_reparto=pd.DataFrame(columns=coche_df.columns)
coche_df_reparto['PESO_TIEMPO']=None
coche_df_reparto['PESO_PRIORIDAD']=None
coche_df_reparto['PESO_DIAS_TRANSCURRIDOS']=None
coche_df_reparto['PESO_TOTAL']=None
```

```
respuesta_coche = str(input('¿Desea programar emplazamientos a los que ir en
coche? (SI/NO)').strip())
```

```
while respuesta_coche!="SI" and respuesta_coche!="NO":
    respuesta_coche = str(input('Escriba una respuesta
válida (SI/NO)').strip())
```

```
while respuesta_coche=="SI":
```

```
    num_paquetes_coche = input('¿Cuántos paquetes de direcciones en coche
quieres?').strip()
```

```
    num_paquetes_coche = int(num_paquetes_coche)
```

```
    emp_coche_trabajador = input('¿A cuántas direcciones en coche va a ir cada
trabajador en su jornada?').strip() #strip hace que no se recojan espacios
blancos
```

```
    emp_coche_trabajador = int(emp_coche_trabajador)
```

```
    for i in range(num_paquetes_coche):
```

```
        direccion_origen=coche_df['DIRECCION'].iloc[0]    #Toma el valor de la
columna 'DIRECCION' fila 0 (primera)
```

```
        clave_primaria_origen=coche_df['CLAVE PRIMARIA'].iloc[0]
```

```
        distancias_desde_primer_punto = []
```

```
        tiempos_desde_primer_punto = []
```

```
        for j in range(1,len(coche_df)):
```

```

    direccion_destino=coche_df['DIRECCION'].iloc[j]

    distancia_coche =
Maps.directions(direccion_origen,direccion_destino + ', Madrid,
Spain',mode='driving')
    km_distancia_coche =
(distancia_coche[0]['legs'][0]['distance']['value']/1000
    min_tiempo_coche =
(distancia_coche[0]['legs'][0]['duration']['value']/60
    km_distancia_coche=round(km_distancia_coche,2)
    min_tiempo_coche=round(min_tiempo_coche,2)

    distancias_desde_primer_punto.append(km_distancia_coche)
    tiempos_desde_primer_punto.append(min_tiempo_coche)

#Creo un dataframe que contendrá los pesos desde la dirección origen
coche_df_pesos = coche_df.drop('DISTANCIA_COCHE',axis=1)
coche_df_pesos = coche_df_pesos.drop('TIEMPO_COCHE',axis=1)
coche_df_pesos=coche_df_pesos.iloc[1:]

coche_df_pesos['DISTANCIA_COCHE']=distancias_desde_primer_punto
coche_df_pesos['TIEMPO_COCHE']=tiempos_desde_primer_punto

#Determino los pesos y los vuelco a dataframe coche_df_pesos
pesos_tiempos=[]
pesos_prioridades=[]
pesos_dias=[]
pesos_totales=[]

min_tiempo=min(tiempos_desde_primer_punto)
max_tiempo=max(tiempos_desde_primer_punto)

dias_transcurridos=coche_df_pesos['DIFERENCIA_DIAS'].dropna().tolist()
min_dias=min(dias_transcurridos)
max_dias=max(dias_transcurridos)

for z in range(len(distancias_desde_primer_punto)):

    peso_tiempo = round(10 - (9/(max_tiempo -
min_tiempo))*(tiempos_desde_primer_punto[z]-min_tiempo),2) # El primer peso
saldrá mayor a 10, pero no va a afectar el resultado
    pesos_tiempos.append(peso_tiempo)

    peso_prioridad = round((10 - coche_df['PRIORIDAD'].iloc[z]),2)
    pesos_prioridades.append(peso_prioridad)
```

```
        peso_dia = round(10 - (9/(max_dias -
min_dias))*(dias_transcurridos[z]-min_dias),2)
        pesos_dias.append(peso_dia)

        peso_total = peso_tiempo + peso_prioridad + peso_dia
        pesos_totales.append(peso_total)

coche_df_pesos['PESO_TIEMPO']=pesos_tiempos
coche_df_pesos['PESO_PRIORIDAD']=pesos_prioridades
coche_df_pesos['PESO_DIAS_TRANSCURRIDOS']=pesos_dias
coche_df_pesos['PESO_TOTAL']=pesos_totales

coche_df_pesos=coche_df_pesos.sort_values(by=['PESO_TOTAL'],ascending=
[False])

for index in range(emp_coche_trabajador):

    if(index==0): #LOS TIEMPOS Y DISTANCIAS DE LA PRIMERA FILA SON
DESDE OFICINA!!

        fila=coche_df.iloc[[index]]          #Para recoger la fila como
un dataframe
        fila.loc[:, 'PESO_TIEMPO'] = [10]
        fila.loc[:, 'PESO_PRIORIDAD'] = [10]
        fila.loc[:, 'PESO_DIAS_TRANSCURRIDOS'] = [10]
        fila.loc[:, 'PESO_TOTAL'] = [30]

        coche_df_reparto=pd.concat([coche_df_reparto,fila],ignore_inde
x=True)

        coche_df=coche_df[coche_df['CLAVE
PRIMARIA']!=clave_primaria_origen] #Se queda con todo el dataframe menos la
que tiene el expediente origen
        coche_df=coche_df.reset_index(drop=True) #Para
reenumerar los índices del dataframe (queda un índice sin contenido tras
borrar)

        df=df[df['CLAVE PRIMARIA']!=clave_primaria_origen]
        df=df.reset_index(drop=True)

    else: #LOS TIEMPOS Y DISTANCIAS RESTANTES SON DESDE LA DIRECCIÓN
DE LA PRIMERA DIRECCIÓN A LA QUE VA EL TRABAJADOR!!

        fila=coche_df_pesos.iloc[[index-1]]

        coche_df_reparto=pd.concat([coche_df_reparto,fila],ignore_inde
x=True)
```

```
clave_primaria=fila['CLAVE PRIMARIA'].values[0] #.values[0]
porque fila es un dataframe con una única fila, por tanto el número de expte
está en el índice cero
```

```
coche_df=coche_df[coche_df['CLAVE
PRIMARIA']!=clave_primaria]
coche_df=coche_df.reset_index(drop=True)
```

```
df=df[df['CLAVE PRIMARIA']!=clave_primaria]
df=df.reset_index(drop=True)
```

```
coche_df_reparto.loc[len(coche_df_reparto)] = [None] *
len(coche_df_reparto.columns)
```

```
respuesta_coche = str(input('¿Desea programar nuevos paquetes de trabajo
en coche? (SI/NO)').strip())
while respuesta_coche!="SI" and respuesta_coche!="NO":
    respuesta_coche = str(input('Escriba una respuesta
válida (SI/NO)').strip())
```

Se general la Excel con los paquetes de trabajo en transporte público y coche, y la Excel con los emplazamientos que quedarían por cubrir:

```
andando_df_reparto['FECHA ALTA'] = andando_df_reparto['FECHA
ALTA'].dt.strftime("%d/%m/%Y")
coche_df_reparto['FECHA ALTA'] = coche_df_reparto['FECHA
ALTA'].dt.strftime("%d/%m/%Y")
```

```
with pd.ExcelWriter('Reparto_generado.xlsx') as writer:
    andando_df_reparto.to_excel(writer,sheet_name='Paquetes_trabajo_andando',i
ndex=False)
    coche_df_reparto.to_excel(writer,sheet_name='Paquetes_trabajo_coche',index
=False)
```

```
with pd.ExcelWriter('Direcciones_final.xlsx') as writer:
    df.to_excel(writer,sheet_name='Direcciones por asistir',index=False)
```

## Anexo II. Amortización del préstamo.

La empresa pide un préstamo de 40.000€ a 36 meses, consiguiendo un interés anual del 6%. Consideraré el sistema de amortización francesa, en el que las cuotas son fijas e incluyen tanto el capital como los intereses. La cuota fija mensual viene dada por:

$$C = \frac{P \cdot r \cdot (1 + r)^n}{(1 + r)^n - 1} = \frac{40.000 \cdot 0,005 \cdot (1 + 0,005)^{36}}{(1 + 0,005)^{36} - 1} = 1.216,88€$$

Donde:

C: cuota mensual

r: interés mensual (6%/12 = 0,5%)

P: importe del préstamo (40.000€)

n: número total de meses (36 meses)

Seguidamente se recoge en forma de tabla la información sobre la cuota mensual, la amortización del capital, los intereses, y el saldo pendiente para cada mes:

Mes	Cuota mensual (€)	Amortización (€)	Intereses (€)	Saldo pendiente (€)
1	1.216,88	1.016,88	200,00	38.983,12
2	1.216,88	1.021,96	194,92	37.961,16
3	1.216,88	1.027,07	189,81	36.934,09
4	1.216,88	1.032,21	184,67	35.901,88
5	1.216,88	1.037,37	179,51	34.864,51
6	1.216,88	1.042,55	174,32	33.821,96
7	1.216,88	1.047,77	169,11	32.774,19
8	1.216,88	1.053,01	163,87	31.721,18
9	1.216,88	1.058,27	158,61	30.662,91
10	1.216,88	1.063,56	153,31	29.599,35
11	1.216,88	1.068,88	148,00	28.530,47
12	1.216,88	1.074,23	142,65	27.456,24
13	1.216,88	1.079,60	137,28	26.376,65
14	1.216,88	1.084,99	131,88	25.291,65
15	1.216,88	1.090,42	126,46	24.201,23
16	1.216,88	1.095,87	121,01	23.105,36
17	1.216,88	1.101,35	115,53	22.004,01
18	1.216,88	1.106,86	110,02	20.897,15
19	1.216,88	1.112,39	104,49	19.784,76
20	1.216,88	1.117,95	98,92	18.666,81
21	1.216,88	1.123,54	93,33	17.543,27
22	1.216,88	1.129,16	87,72	16.414,10
23	1.216,88	1.134,81	82,07	15.279,30
24	1.216,88	1.140,48	76,40	14.138,82

Mes	Cuota mensual (€)	Amortización (€)	Intereses (€)	Saldo pendiente (€)
25	1.216,88	1.146,18	70,69	12.992,63
26	1.216,88	1.151,91	64,96	11.840,72
27	1.216,88	1.157,67	59,20	10.683,05
28	1.216,88	1.163,46	53,42	9.519,58
29	1.216,88	1.169,28	47,60	8.350,30
30	1.216,88	1.175,13	41,75	7.175,18
31	1.216,88	1.181,00	35,88	5.994,18
32	1.216,88	1.186,91	29,97	4.807,27
33	1.216,88	1.192,84	24,04	3.614,43
34	1.216,88	1.198,81	18,07	2.415,62
35	1.216,88	1.204,80	12,08	1.210,82
36	1.216,88	1.210,82	6,05	0,00

*Tabla 12.- Amortizaciones e intereses mensuales del préstamo pedido por EcoRutas el primer año. Fuente: elaboración propia.*

## Anexo III. Tablas con los flujos de caja de los tres primeros años.

Mes	Año 2026												
	0	1	2	3	4	5	6	7	8	9	10	11	12
Capital	80.000,00												
Crédito		40.000,00											
Cobros clientes						2.312,50	4.625,00	6.937,50	9.250,00	11.562,50	13.875,00	16.187,50	18.500,00
<b>Total ingresos</b>	<b>80.000,00</b>	<b>40.000,00</b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>	<b>2.312,50</b>	<b>4.625,00</b>	<b>6.937,50</b>	<b>9.250,00</b>	<b>11.562,50</b>	<b>13.875,00</b>	<b>16.187,50</b>	<b>18.500,00</b>
Inversión equipos informáticos					-7.005,00								
Subcontratación interfaz		-1.500,00	-1.500,00	-1.500,00									
Auditoría SEO					-1.500,00								
Marketing y publicidad					-6.000,00	-6.000,00	-6.000,00	-6.000,00					
Costes salariales		-2.800,00	-2.800,00	-2.800,00	-10.600,00	-10.600,00	-10.600,00	-10.600,00	-10.600,00	-10.600,00	-10.600,00	-10.600,00	-10.600,00
Gastos operativos IT y comunicaciones		-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00
Amortización préstamo		-1016,877	-1021,962	-1027,071695	-1032,207053	-1037,368089	-1042,555	-1047,768	-1053,007	-1058,272	-1063,563	-1068,8807	-1074,2252
Intereses		-200	-194,9156	-189,8058031	-184,6704446	-179,5094093	-174,3226	-169,1098	-163,871	-158,6059	-153,3146	-147,99675	-142,65235
<b>Total pagos</b>		<b>-6.228,88</b>	<b>-6.228,88</b>	<b>-6.228,88</b>	<b>-27.033,88</b>	<b>-18.528,88</b>	<b>-18.528,88</b>	<b>-18.528,88</b>	<b>-12.528,88</b>	<b>-12.528,88</b>	<b>-12.528,88</b>	<b>-12.528,88</b>	<b>-12.528,88</b>
<b>Variación mensual caja</b>	<b>80.000,00</b>	<b>33.771,12</b>	<b>-6.228,88</b>	<b>-6.228,88</b>	<b>-27.033,88</b>	<b>-16.216,38</b>	<b>-13.903,88</b>	<b>-11.591,38</b>	<b>-3.278,88</b>	<b>-966,38</b>	<b>1.346,12</b>	<b>3.658,62</b>	<b>5.971,12</b>
<b>Acumulado en caja</b>	<b>80.000,00</b>	<b>113.771,12</b>	<b>107.542,25</b>	<b>101.313,37</b>	<b>74.279,49</b>	<b>58.063,11</b>	<b>44.159,24</b>	<b>32.567,86</b>	<b>29.288,98</b>	<b>28.322,60</b>	<b>29.668,73</b>	<b>33.327,35</b>	<b>39.298,47</b>

Tabla 13.- Flujos de caja de EcoRutas esperados para el año 2026. Fuente: elaboración propia.

Mes	Año 2027											
	1	2	3	4	5	6	7	8	9	10	11	12
Cobros clientes	20.812,50	23.125,00	25.437,50	27.750,00	30.756,25	33.762,50	36.768,75	39.775,00	42.781,25	45.787,50	48.793,75	51.800,00
<b>Total ingresos</b>	<b>20.812,50</b>	<b>23.125,00</b>	<b>25.437,50</b>	<b>27.750,00</b>	<b>30.756,25</b>	<b>33.762,50</b>	<b>36.768,75</b>	<b>39.775,00</b>	<b>42.781,25</b>	<b>45.787,50</b>	<b>48.793,75</b>	<b>51.800,00</b>
Inversión equipos informáticos	-2.335,00											
Costes salariales	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00	-18.600,00
Bonificaciones por desempeño	-11.512,50											
Gastos operativos IT y comunicaciones	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00
Amortización préstamo	-1079,596	-1084,994	-1090,419	-1095,871326	-1101,350683	-1106,857436	-1112,392	-1117,954	-1123,543	-1129,161	-1134,807	-1140,481
Intereses	-137,28	-131,88	-126,46	-121,01	-115,53	-110,02	-104,49	-98,92	-93,33	-87,72	-82,07	-76,40
Impuestos								0,00				
Dividendos												-40.000,00
<b>Total pagos</b>	<b>-34.376,38</b>	<b>-20.528,88</b>	<b>-60.528,88</b>									
<b>Variación mensual caja</b>	<b>-13.563,88</b>	<b>2.596,12</b>	<b>4.908,62</b>	<b>7.221,12</b>	<b>10.227,37</b>	<b>13.233,62</b>	<b>16.239,87</b>	<b>19.246,12</b>	<b>22.252,37</b>	<b>25.258,62</b>	<b>28.264,87</b>	<b>-8.728,88</b>
<b>Acumulado caja</b>	<b>25.734,59</b>	<b>28.330,72</b>	<b>33.239,34</b>	<b>40.460,46</b>	<b>50.687,83</b>	<b>63.921,46</b>	<b>80.161,33</b>	<b>99.407,45</b>	<b>121.659,82</b>	<b>146.918,45</b>	<b>175.183,32</b>	<b>166.454,44</b>

Tabla 14.- Flujos de caja de EcoRutas esperados para el año 2027. Fuente: elaboración propia.

	Año 2028											
Mes	1	2	3	4	5	6	7	8	9	10	11	12
Cobros clientes	54.806,25	57.812,50	60.818,75	63.825,00	65.212,50	66.600,00	67.987,50	69.375,00	70.762,50	72.150,00	73.537,50	74.925,00
<b>Total ingresos</b>	<b>54.806,25</b>	<b>57.812,50</b>	<b>60.818,75</b>	<b>63.825,00</b>	<b>65.212,50</b>	<b>66.600,00</b>	<b>67.987,50</b>	<b>69.375,00</b>	<b>70.762,50</b>	<b>72.150,00</b>	<b>73.537,50</b>	<b>74.925,00</b>
Inversión equipos informáticos	-7.005,00											
Costes salariales	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00	-31.800,00
Bonificaciones por desempeño	-24.750,00											
Gastos operativos IT y comunicaciones	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00	-712,00
Amortización préstamo	-1.146,18	-1.151,91	-1.157,67	-1.163,46	-1.169,28	-1.175,13	-1.181,00	-1.186,91	-1.192,84	-1.198,81	-1.204,80	-1.210,82
Intereses	-70,69	-64,96	-59,20	-53,42	-47,60	-41,75	-35,88	-29,97	-24,04	-18,07	-12,08	-6,05
Impuestos	-14.098,03											
Dividendos												-160.000,00
<b>Total pagos</b>	<b>-79.581,91</b>	<b>-33.728,88</b>	<b>-193.728,88</b>									
<b>Variación mensual caja</b>	<b>-24.775,66</b>	<b>24.083,62</b>	<b>27.089,87</b>	<b>30.096,12</b>	<b>31.483,62</b>	<b>32.871,12</b>	<b>34.258,62</b>	<b>35.646,12</b>	<b>37.033,62</b>	<b>38.421,12</b>	<b>39.808,62</b>	<b>-118.803,88</b>
<b>Acumulado caja</b>	<b>141.678,78</b>	<b>165.762,40</b>	<b>192.852,28</b>	<b>222.948,40</b>	<b>254.432,02</b>	<b>287.303,14</b>	<b>321.561,77</b>	<b>357.207,89</b>	<b>394.241,51</b>	<b>432.662,63</b>	<b>472.471,26</b>	<b>353.667,38</b>

Tabla 15.- Flujos de caja de EcoRutas esperados para el año 2028. Fuente: elaboración propia.

## Anexo IV. Asientos contables de los tres primeros años.

En la realización de los asientos contables, que se recogen en este anexo, se han tenido en cuenta las siguientes consideraciones:

- Las comisiones por venta que se llevan plataformas como GitHub o Gumroad están directamente vinculadas a la transacción de un producto o servicio específico. Se ha optado por considerarlas como CMV, ya que estas comisiones solo ocurren cuando se realiza una venta, son costos variables asociados directamente a cada transacción y forman parte de los costos necesarios para entregar el producto al cliente a través de dichas plataformas.
- Por simplicidad, la interfaz de usuario se ha asumido que no es capitalizable, por lo que se ha tenido en cuenta como un gasto operativo asociado a la subcontratación de un servicio profesional.
- La auditoría SEO es una actividad aislada, destinada a analizar y optimizar el sitio web, pero sin y se registra directamente en la cuenta de resultados como un gasto operativo.
- En España, la Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades recoge en el art. 29.2 establece un tipo reducido del 15% para nuevas empresas en el primer período impositivo en que la base imponible resulte positiva y en el siguiente. Según el art. 29.1 con carácter general, el tipo de gravamen es del 25% para los contribuyentes de este Impuesto.
- El primer año la base imponible resulta negativa, por lo que la empresa no deberá pagar el impuesto de sociedades.

### ASIENTOS CONTABLES AÑO 2026

<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Caja y bancos</div> 80.000 (1) Constitución de la empresa 40.000 (2) Crédito a tres años 90.000 (3) Ventas -6.750 (4) CMV (comisión plataformas de venta) -7.005 (5) Equipos informáticos -4.500 (6) Subcontrata interfaz -1.500 (7) Auditoría SEO -24.000 (8) Marketing y publicidad -8.544 (9) IT y comunicaciones -103.800 (10) Coste empresa trabajadores -12.543,76 (13) Amortización préstamo -2.058,77 (14) Intereses préstamo <b>39.298,47</b>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Activo inmovilizado</div> 7.005 (5) Equipos informáticos -1.050,75 (12) Amortización equipos informáticos <b>5.954,25</b>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Socios</div> 80.000 (1) Constitución de la empresa <b>80.000</b>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Reservas</div> <b>-73.716,02</b> (16) Reservas
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Deuda a largo</div> 40.000 (2) Crédito a tres años -12.543,76 (13) Amortización préstamo <b>27.456,24</b>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Cuenta de resultados</div> 90.000 (3) Ventas -6.750 (4) CMV (comisión plataformas de venta) -4.500 (6) Subcontrata interfaz -1.500 (7) Auditoría SEO -24.000 (8) Marketing y publicidad -8.544 (9) IT y comunicaciones -103.800 (10) Coste empresa trabajadores -11.512,50 (11) Devengo bonos 2026 -1.050,75 (12) Amortización equipos informáticos -2.058,77 (14) Intereses préstamo 0,00 (15) Impuestos 2026* 73.716,02 (16) Reservas <b>0,00</b>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Impuestos</div> 0,00 (15) Impuestos 2026* <b>0,00</b>	
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Bonos por pagar</div> 11.512,50 (11) Devengo bonos 2026 <b>11.512,50</b>	

Fig. 28.- : Asientos contables del año 2026. Fuente: elaboración propia.

## ASIENTOS CONTABLES 2027

<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Caja y bancos</div> <p style="margin-left: 20px;">39.298,47</p> <p style="margin-left: 20px;">462.000,00 (1) Ventas</p> <p style="margin-left: 20px;">-34.650,00 (2) CMV (comisión plataformas de venta)</p> <p style="margin-left: 20px;">-2.335,00 (3) Equipos informáticos</p> <p style="margin-left: 20px;">-11.512,50 (4) Pago bonos 2025</p> <p style="margin-left: 20px;">-8.544,00 (5) IT y comunicaciones</p> <p style="margin-left: 20px;">-223.200,00 (6) Coste empresa trabajadores</p> <p style="margin-left: 20px;">-13.317,43 (8) Amortización préstamo</p> <p style="margin-left: 20px;">-1.285,10 (10) Intereses préstamo</p> <p style="margin-left: 20px;">-40.000 (13) Dividendos</p> <p style="margin-left: 20px;">166.454,44</p>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Activo inmovilizado</div> <p style="margin-left: 20px;">5.954,25</p> <p style="margin-left: 20px;">2.335,00 (3) Equipos informáticos</p> <p style="margin-left: 20px;">-1.868,00 (7) Amortización equipos informáticos</p> <p style="margin-left: 20px;">6.421,25</p>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Socios</div> <p style="margin-left: 20px;">80.000</p>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Reservas</div> <p style="margin-left: 20px;">-73.716,02</p> <p style="margin-left: 20px;">113.604,87 (12) Reservas</p> <p style="margin-left: 20px;">39.888,84</p>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Deuda a largo</div> <p style="margin-left: 20px;">27.456,24</p> <p style="margin-left: 20px;">-13.317,43 (8) Amortización préstamo</p> <p style="margin-left: 20px;">14.138,81</p>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Cuenta de resultados</div> <p style="margin-left: 20px;">462.000,00 (1) Ventas</p> <p style="margin-left: 20px;">-34.650,00 (2) CMV</p> <p style="margin-left: 20px;">-8.544,00 (5) IT y comunicaciones</p> <p style="margin-left: 20px;">-223.200,00 (6) Coste empresa trabajadores</p> <p style="margin-left: 20px;">-1.868,00 (7) Amortización equipos informáticos</p> <p style="margin-left: 20px;">-24.750,00 (9) Devengo bonos 2027</p> <p style="margin-left: 20px;">-1.285,10 (10) Intereses préstamo</p> <p style="margin-left: 20px;">-14.098,03 (11) Impuestos 2027</p> <p style="margin-left: 20px;">-113.604,87 (12) Reservas</p> <p style="margin-left: 20px;">-40.000 (13) Dividendos</p> <p style="margin-left: 20px;">0,00</p>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Impuestos</div> <p style="margin-left: 20px;">0,00</p> <p style="margin-left: 20px;">14.098,03 (11) Impuestos 2027</p> <p style="margin-left: 20px;">14.098,03</p>	
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Bonos por pagar</div> <p style="margin-left: 20px;">11.512,50</p> <p style="margin-left: 20px;">-11.512,50 (5) Pago bonos 2026</p> <p style="margin-left: 20px;">24.750,00 (9) Devengo bonos 2027</p> <p style="margin-left: 20px;">24.750,00</p>	

Fig. 29.- Asientos contables del año 2027. Fuente: elaboración propia.

## ASIENTOS CONTABLES 2028

<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Caja y bancos</div> <p>166.454,44 -14.098,03 (1) Impuestos 2027 862.500,00 (2) Ventas -64.687,50 (3) CMV (comisión plataformas de venta) -7.005,00 (4) Equipos informáticos -24.750,00 (5) Pago bonos 2027 -8.544,00 (6) IT y comunicaciones -381.600,00 (7) Coste empresa trabajadores -14.138,82 (9) Amortización préstamo -463,71 (11) Intereses préstamo -160.000 (14) Dividendos 353.667,37</p>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Activo inmovilizado</div> <p>6.421,25 7.005,00 (4) Equipos informáticos -3.269,00 (8) Amortización equipos informáticos 10.157,25</p>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Socios</div> <p>80.000</p>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Reservas</div> <p>39.888,84 147.390,42 (13) Reservas 187.279,26</p>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Deuda a largo</div> <p>14.138,81 -14.138,82 (9) Amortización préstamo 0,00</p>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Cuenta de resultados</div> <p>862.500,00 (2) Ventas -64.687,50 (3) CMV -8.544,00 (6) IT y comunicaciones -381.600,00 (7) Coste empresa trabajadores -3.269,00 (8) Amortización equipos informáticos -42.300,00 (10) Devengo bonos 2028 -463,71 (11) Intereses préstamo -54.245,37 (12) Impuestos 2028 -147.390,42 (13) Reservas -160.000 (14) Dividendos 0,00</p>
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Impuestos</div> <p>14.098,03 -14.098,03 (1) Impuestos 2027 54.245,37 (12) Impuestos 2028</p>	
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Bonos por pagar</div> <p>24.750,00 -24.750,00 (5) Pago bonos 2027 42.300,00 (10) Devengo bonos 2028 42.300,00</p>	

Fig. 30.- Asientos contables del año 2028. Fuente: elaboración propia.

# BIBLIOGRAFÍA

## Capítulo 1. Introducción.

Rodríguez-Molina, J., & Sánchez-Moreno, J. (2019). Desafíos logísticos y sostenibilidad en la era de las restricciones urbanas. *Revista Española de Logística*, 29(1), 34-49.

Alonso, L., Rodríguez, J., & Torres, M. (2021). Gestión logística y sostenibilidad: Una perspectiva para pequeñas y medianas empresas. *Journal of Transport & Environment*, 12(4), 267-282.

## Capítulo 2. Estado del arte de los softwares para optimización de rutas.

Baptista, S., Oliveira, R. C., & Zúquete, E. (2019). Vehicle Routing Problems: An Overview of the State-of-the-Art and Future Research Directions. *Transportation Research Procedia*, 39, 244–252. <https://www.researchgate.net/>

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. <https://ir.cwi.nl/>

Albalade, D. (2021, agosto 3). Algoritmo Dijkstra con giros permitidos. Medium. <https://medium.com/>

## Capítulo 3. Desarrollo y explicación del código de optimización de rutas generado.

Python Software Foundation. (2024). Python Documentation. <https://docs.python.org/>

NetworkX Developers. (2024). NetworkX Documentation. <https://networkx.org/documentation/stable/>

Scipy Community. (2024). Scipy Documentation. <https://docs.scipy.org/>

Python Software Foundation. (n.d.). *Documentación de la biblioteca estándar de Python*. <https://docs.python.org/es/3/library/index.html>

pandas Development Team. (n.d.). *User guide — pandas documentation*. [https://pandas.pydata.org/docs/user\\_guide](https://pandas.pydata.org/docs/user_guide)

Python Software Foundation. (n.d.). *Módulo datetime — Python 3.13.0 documentation*. <https://docs.python.org/es/3.13/library/datetime.html#>

Google Cloud. (n.d.). *Características gratuitas de Google Cloud*. <https://cloud.google.com/free/docs/free-cloud-features?hl=es-419>

Maplink. (n.d.). *Cómo obtener una Google Maps API Key*. <https://blog.maplink.global/es/como-obtener-google-maps-api-key/>

Google Developers. (n.d.). *Directions API*. <https://developers.google.com/maps/documentation/directions/>

OpenRouteService. (n.d.). *Planes de OpenRouteService*. <https://openrouteservice.org/plans/>

#### **Capítulo 4. Plan de negocio.**

Poyatos Díaz, J. M. (2024). *Nota técnica para los alumnos del Máster de Ingeniería Industrial de la Universidad Europea de Madrid, curso 2024/25*. Copyright Juan Miguel Poyatos Díaz 2024.

Lacort Peña, B. (2024-2025). Apuntes de la asignatura Dirección de grandes proyectos. Universidad Europea de Madrid

Instituto Nacional de Estadística (INE). (n.d.). *Estadística Estructural de Empresas: Sector Servicios Año 2022* <https://www.ine.es/dyngs/Prensa/es/EEESS2022.htm>

Mordor Intelligence. (n.d.). *Mercado de software de optimización de rutas*. <https://www.mordorintelligence.com/es/industry-reports/route-optimization-software-market>

Infinitive Data Expert. (n.d.). *Informe del mercado de software de optimización de rutas*. <https://www.infinitivedataexpert.com/es/industry-report/route-optimization-software-market>

Canva. (n.d.). *Plantillas*. [https://www.canva.com/es\\_es/plantillas/](https://www.canva.com/es_es/plantillas/)

LanRoute. "Cuadrante Mágico de Gartner". <https://lanroute.com>

Route4Me. (n.d.). Planificación y optimización de rutas. <https://route4me.com>

Gumroad. (n.d.). Sell what you know and see what sticks. <https://gumroad.com/>

GitHub. (n.d.). Where the world builds software. <https://github.com/>

HERE Technologies. (n.d.). Pricing plans for location services. <https://www.here.com/get-started/pricing>

Google Ads. (n.d.). *Publicidad en línea efectiva con Google Ads*. [https://ads.google.com/intl/es\\_es/home/](https://ads.google.com/intl/es_es/home/)

LinkedIn. (n.d.). *LinkedIn Ads: Soluciones de marketing para empresas*. <https://business.linkedin.com/es-es/marketing-solutions/cx/21/12/ads>

MeaningCloud. (n.d.). *Voz del empleado, voz del cliente y NPS*. <https://www.meaningcloud.com/es/blog/voz-del-empleado-voz-del-cliente-y-nps>

Impulsa Empresa. (n.d.). *Coste del trabajador para la empresa*. <https://www.impulsa-empresa.es/coste-trabajador-empresa/>

Intuit QuickBooks. (n.d.). *QuickBooks Global: Software de contabilidad y gestión financiera*. <https://quickbooks.intuit.com/global/es-es/>

Microsoft. (n.d.). *Windows 11 Pro para empresas*. <https://www.microsoft.com/es-es/windows/business/windows-11-pro>

Microsoft. (n.d.). *Comprar Microsoft 365*. <https://www.microsoft.com/es-ES/microsoft-365/buy/>

CloudEndure. (n.d.). *CloudEndure: Soluciones de recuperación ante desastres*. <https://www.cloudendure.com/>

Google Cloud. (n.d.). *Google Cloud: Soluciones en la nube*. <https://cloud.google.com/?hl=es>

España. (2014). *Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades*. Boletín Oficial del Estado (BOE), núm. 288, de 28 de noviembre de 2014, pp. 96939-97058. <https://www.boe.es/buscar/doc.php?id=BOE-A-2014-12328>