



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

**Visualiza: procesamiento y visualización de
datos abiertos sobre el empleo en Canarias**

*Visualiza: processing and visualization of open data on
employment in the Canary Islands*

Dayana Armas Alonso

La Laguna, 13 de septiembre de 2022

D. **Francisco Javier Rodríguez González**, con N.I.F. 43.618.712-V profesor Asociado de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Alejandro Pérez Nava**, con N.I.F. 43.821.179-S profesor Asociado de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Visualiza: procesamiento y visualización de datos abiertos sobre el empleo en Canarias"

ha sido realizada bajo su dirección por D. **Dayana Armas Alonso**, con N.I.F. 79.151.932-P.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de septiembre de 2022

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia por apoyarme durante todo mi camino. También me gustaría agradecer a mis amigos, incluidas a todas las personas que he conocido durante la carrera y me han apoyado incluso en los momentos más duros. Y, finalmente, me gustaría agradecer a mi tutor Francisco por ayudarme, apoyarme y guiarme para poder llevar a cabo este proyecto. Estaré eternamente agradecida por el apoyo en estos cuatro años de carrera y por formar parte de mi crecimiento. No habría llegado hasta aquí sin ninguno de ellos.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial-SinObraDerivada 4.0 Internacional.

Resumen

En los últimos años, las organizaciones han convertido en un activo crucial el Big Data debido a la gran variedad de conjuntos de datos que se generan en grandes volúmenes y a una velocidad cada vez mayor. Estos conjuntos de datos llegan a ser tan voluminosos que el software de procesamiento de datos convencional sencillamente no puede gestionarlos. Por ello, nace el concepto de visualización de datos que consiste en representar grandes conjuntos de datos en información enriquecida. Los datos sobre el empleo que generan las administraciones públicas en las Islas Canarias son realmente importantes debido a la situación laboral de los ciudadanos que influye en la economía del país. Es por ello, que surge la necesidad de mostrar estos datos de forma intuitiva dada la gran cantidad de datos que se han ido almacenando hasta la actualidad.

El siguiente proyecto consiste en desarrollar una aplicación en versión web y en versión móvil que permita analizar, procesar y visualizar datos abiertos, es decir, conjuntos de datos que son de uso público y se pueden utilizar libremente. Para ello, los conjuntos de datos en relación al empleo se extraerán a través de la API de Canarias Datos Abiertos. Con la representación de estos datos, es posible observar la evolución de la economía en Canarias apreciando ciertos patrones que pudiesen mejorar la tasa de empleo y a su vez, facilitar la toma de decisiones a las empresas y organismos públicos. Con ello, se podría llevar a cabo una transformación del modelo económico de Canarias y de esta forma, no llegar a ser tan dependientes del sector turístico.

Además de desarrollar esta aplicación, se ha llevado a cabo un estudio de la viabilidad del proyecto teniendo una duración estimada de 9 meses, y con un retorno de inversión factible que se obtendría a los 22 meses y 2 semanas desde la comercialización. Por lo que, sería una aplicación potente con grandes beneficios que tendría una gran escalabilidad ya que se podría ampliar el catálogo de los conjuntos de datos y recoger información de otras provincias de España, e incluso se podría realizar gráficas comparativas entre distintos países, aumentando de esta forma, el interés de los usuarios.

Palabras clave: Big Data, conjuntos de datos, visualización de datos, Islas Canarias

Abstract

In recent years, organizations have made Big Data a crucial asset due to the wide variety of data sets being generated in large volumes and at ever-increasing speed. These data sets become so large that conventional data processing software simply cannot handle them. For this reason, the concept of data visualization was born, which consists of representing large data sets in enriched information. The data on employment generated by public administrations in the Canary Islands are really important due to the employment situation of citizens that influences the country's economy. That is why the need arises to show this data intuitively given the large amount of data that has been stored up to now.

The following project consists of developing an application in a web version and in a mobile version that allows analyzing, processing and visualizing open data, that is, data sets that are for public use and can be used freely. To do this, the data sets in relation to employment will be extracted through the Canarias Open Data API. With the representation of these data, it is possible to observe the evolution of the economy in the Canary Islands, appreciating certain patterns that could improve the employment rate and, in turn, facilitate decision-making for companies and public bodies. With this, a transformation of the economic model of the Canary Islands could be carried out and in this way, not become so dependent on the tourism sector.

In addition to developing this application, a feasibility study of the project has been carried out with an estimated duration of 9 months, and with a feasible return on investment that would be obtained at 22 months and 2 weeks from commercialization. Therefore, it would be a powerful application with great benefits that would have great scalability since the catalog of data sets could be expanded and information from other provinces of Spain could be collected, and comparative graphs could even be made between different countries, increased from this way, the interest of the users.

Keywords: Big Data, datasets, data visualization, Canary Islands

Índice general

1. Introducción	1
1.1. Definición del problema	1
1.2. Justificación	3
1.3. Tendencia del mercado	4
1.4. Objetivos	6
2. Estudio previo	8
2.1. Lenguaje de programación y frameworks	8
2.2. Entorno de desarrollo	11
2.3. Elección de tecnologías y entorno de desarrollo	13
2.4. Dificultades encontradas en las tecnologías	14
3. Desarrollo de arquitecturas	15
3.1. Procesamiento de datos	15
3.1.1. Extracción, transformación y carga (ETL)	15
3.1.2. Creación de nuevos conjuntos de datos	17
3.1.3. Actualización de conjuntos de datos	17
3.2. Estructura backend	18
3.3. Estructura librería de visualización de datos	21
3.4. Estructura base de datos	24
3.5. Estructura frontend	25
3.5.1. Frontend web	26
3.5.2. Frontend móvil	28
3.6. Despliegue	31
3.6.1. Entorno Heroku	31
3.6.2. Entorno Android Studio	32
4. Estudio de viabilidad económica	33
4.1. Desarrollo del proyecto	33
4.2. Funcionalidades adicionales	36
4.3. Modelo de comercialización	36
4.4. Punto de Retorno de Inversión (ROI)	37
5. Conclusiones y líneas futuras	40
6. Summary and Conclusions	41
7. Presupuesto	42

Índice de Figuras

1.1. Cantidad real y prevista de datos generados en todo el mundo [1]	1
1.2. Usos del Big Data en el sector financiero [2]	2
1.3. Evolución mensual del número de desempleados en Canarias [3]	2
1.4. Evolución del PIB en Canarias [4]	3
1.5. Evolución de ingresos del tamaño del mercado de Big Data en todo el mundo[5]	4
2.1. Logos de node.js con Express.js y Spring Boot	8
2.2. Logos de PostgreSQL y MongoDB	9
2.3. Logos de ReactJS, Angular y VueJS	10
2.4. Logos de React Native e Ionic	10
2.5. Logos de D3.js y Chart.js	11
2.6. Logos de GitHub, Visual Studio Code y Android Studio	12
2.7. Logos de Heroku y Thunder Client	12
2.8. Logos de MongoDB Atlas y pgAdmin	12
2.9. Porcentaje de dudas de frameworks más preguntados durante los años [6]	13
3.1. Uso del Thunder Client para obtener el id de origen del paquete	16
3.2. Uso del Thunder Client para obtener la url que devuelve el contenido de datos	16
3.3. Estructura de cadena para representar el periodo de node-cron [7]	19
3.4. Componente Pie de la librería react-chartjs-2	21
3.5. Gráfica de la aplicación de tipo Bar Chart	21
3.6. Gráfica de la aplicación de tipo Pie Chart	22
3.7. Gráfica de la aplicación de tipo Line Chart	22
3.8. Gráfica de la aplicación de tipo Polar Area Chart	23
3.9. Gráfica de la aplicación de tipo Radar Chart	23
3.10 Gráfica de la aplicación que combina los tipos Bar Chart y Line Chart	24
3.11 Ejemplo de conjunto de datos almacenado (1)	25
3.12 Ejemplo de conjunto de datos almacenado (2)	25
3.13 Página principal del dashboard de la aplicación web (1)	26
3.14 Página principal del dashboard de la aplicación web (2)	27
3.15 Página principal Dashboard de la aplicación web (3)	27
3.16 Sección de Poblaciones de la aplicación (1)	28
3.17 Sección de Poblaciones de la aplicación (2)	28
3.18 Página principal del dashboard de la aplicación móvil	29
3.19 Filtro desplegado de la sección de Poblaciones de la aplicación móvil	30
3.20 Sección de Poblaciones de la aplicación móvil	30
3.21 Vista horizontal de la aplicación móvil	31
4.1. Diagrama de Gantt (1)	34
4.2. Diagrama de Gantt (2)	34

4.3. Cálculos del Punto de Retorno de Inversión (1)	38
4.4. Cálculos del Punto de Retorno de Inversión (2)	38
4.5. Cálculos del Punto de Retorno de Inversión (3)	38
4.6. Punto de Retorno de Inversión	39

Índice de Tablas

7.1. Presupuesto del proyecto de Trabajo de Fin de Grado 42

Capítulo 1

Introducción

En este primer capítulo se define el problema sobre la necesidad de visualizar datos acerca del empleo y además, se justifica y se compara entre otras aplicaciones existentes del mercado. Finalmente, se plantean los objetivos que se han tenido en cuenta para elaborar este proyecto.

1.1. Definición del problema

Durante los últimos años, se ha incrementado rápidamente la cantidad de datos que recopilan y almacenan las empresas. En la siguiente gráfica se puede mostrar la evolución del Big Data ¹ y además, se observa una predicción de los siguientes años.



Figura 1.1: Cantidad real y prevista de datos generados en todo el mundo [1]

La importancia del Big Data no gira entorno al crecimiento de los datos sino lo que se puede llegar a hacer con ellos. Es por ello que en la siguiente gráfica se puede mostrar los porcentajes de varios usos del Big Data en el sector financiero, destacando la necesidad de las empresas en manejar adecuadamente el uso de los datos.

¹**Big Data:** es un término que hace referencia a los datos que son tan grandes, rápidos o complejos que es difícil o imposible procesarlos con los métodos tradicionales.

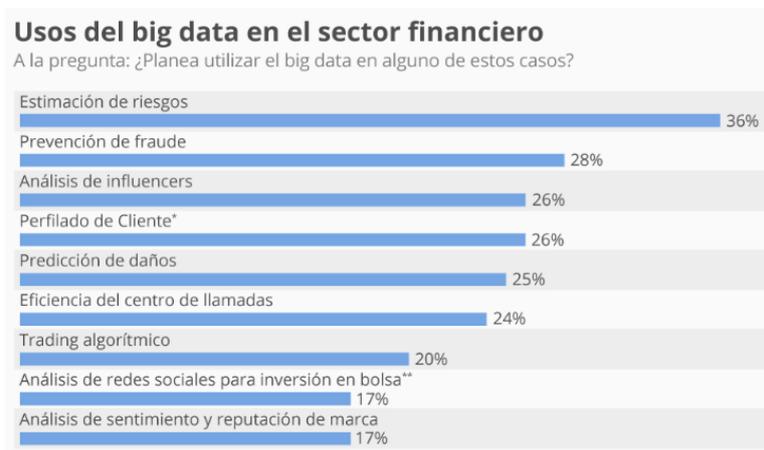


Figura 1.2: Usos del Big Data en el sector financiero [2]

Por lo que, el crecimiento de los datos y el gran interés para hacer un gran uso de los mismos, ha generado ciertos problemas a la hora de analizar los conjuntos de datos ya que se dificulta la extracción de la información útil debido a la falta de comprensión.

Por ello nace el campo interdisciplinario de ciencia de datos que convierte los datos en conocimiento útil involucrando métodos científicos, procesos y sistemas para extraer conocimiento o un mejor entendimiento de los datos en sus diferentes formas, ya sean estructurados o no estructurados. A través del procesamiento y la manipulación de estos, se consigue realizar análisis avanzados que procesan de manera eficiente la información, obteniendo mejores decisiones y movimientos estratégicos de los negocios.

La visualización de datos es uno de los pasos del proceso de la ciencia de datos que permite obtener el valor de los datos a través de la representación gráfica de la información empleando diferentes tipos de herramientas visuales, como mapas, gráficos o infografías, que ayudan a presentar la información de manera visual, permitiendo que sea más accesible para el usuario y reconociendo relevantes valores, tendencias, patrones infrecuentes, etc. La visualización de datos en el ámbito del empleo es muy importante debido a la situación actual que se está viviendo en Canarias donde se ha visto el crecimiento del desempleo ocasionado por la pandemia mundial de la COVID-19.

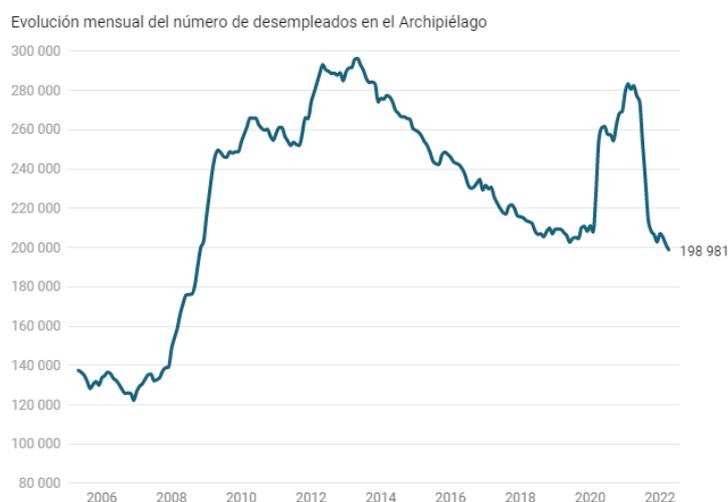
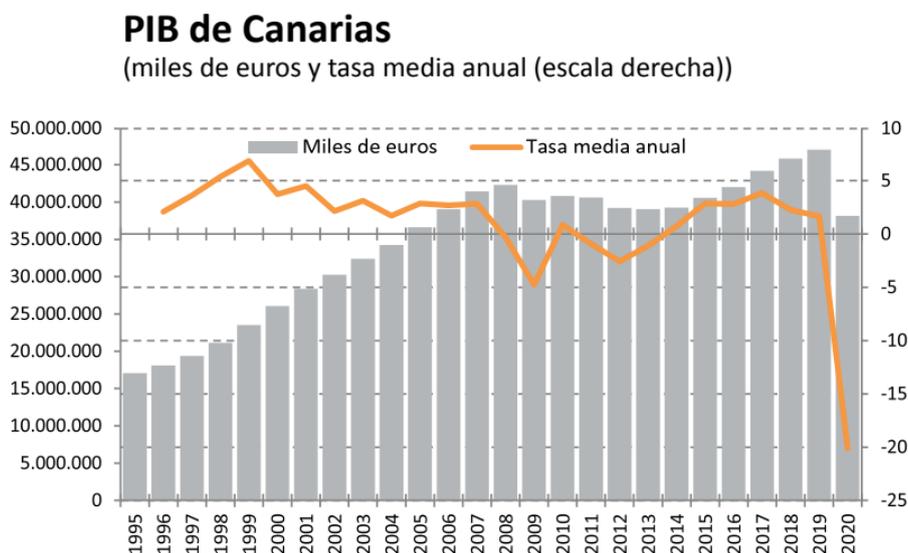


Figura 1.3: Evolución mensual del número de desempleados en Canarias [3]

Esto se ve reflejado ya que el sector con gran importancia que abastece la economía de Canarias es el sector turístico y debido a la situación anterior, muchos turistas no pudieron visitar las islas ya que se cerraron los aeropuertos, provocando un decrecimiento en la economía del archipiélago que, como consecuencia, ha generado un crecimiento en la tasa de desempleo.

Esto se puede ver reflejado en la siguiente gráfica donde se observa el decrecimiento del PIB ² durante el comienzo de la pandemia.



Fuente: Elaboración CEOE-Tenerife/Corporación 5.ISTAC, Contabilidad Regional de España (CRE). (Base 2015)
Nota: En rojo, series anteriores al cambio de base en 2000. El dato de 2020 es el avance que proporciona la Contabilidad Trimestral de Canarias. En ediciones anteriores se utilizó la CRE con base 2010.

Figura 1.4: Evolución del PIB en Canarias [4]

Por lo tanto, este proyecto de aplicación pretende ayudar a mejorar esta situación a través de la ciencia de datos. Para ello, se encargará de extraer, procesar y visualizar conjuntos de datos abiertos ³ sobre el empleo en Canarias, a través de la API⁴ de Canarias Datos Abiertos [8]. Con ello, será posible ver factores que afecten al empleo y de esta manera, mejorar la situación económica de las islas.

1.2. Justificación

Esta aplicación permitirá comprender rápidamente los datos de forma visual a través del uso de las visualizaciones, aumentando significativamente la velocidad de los procesos de toma de decisiones en las empresas y organismos públicos.

Además, permitirá conocer tendencias o patrones que ayuden a mejorar la economía

²**PIB:** trata de un conjunto de los bienes y servicios producidos en un país durante un espacio de tiempo.

³**Datos abiertos:** es una filosofía y práctica que persigue que determinados datos estén disponibles de forma libre a todo el mundo, sin restricciones de copyright, patentes u otros mecanismos de control.

⁴**API:** es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

en Canarias, logrando una transformación del modelo económico⁵ y de esta forma, no llegar a ser tan dependientes del sector turístico.

Actualmente en el mercado no existe una aplicación que procese y trate los conjuntos de datos de la API de Canarias Datos Abiertos. Por tanto, el presente trabajo de investigación académica pretende realizar un estudio de herramientas y tecnologías de última generación y a futuro, con las que desarrollar un prototipo que permita cubrir las necesidades actuales del mercado en relación al Big Data.

1.3. Tendencia del mercado

En la tendencia actual del mercado se observa que la cantidad de dinero invertido entorno al Big Data se encuentra en constante crecimiento debido a la necesidad de manejar grandes conjuntos de datos.

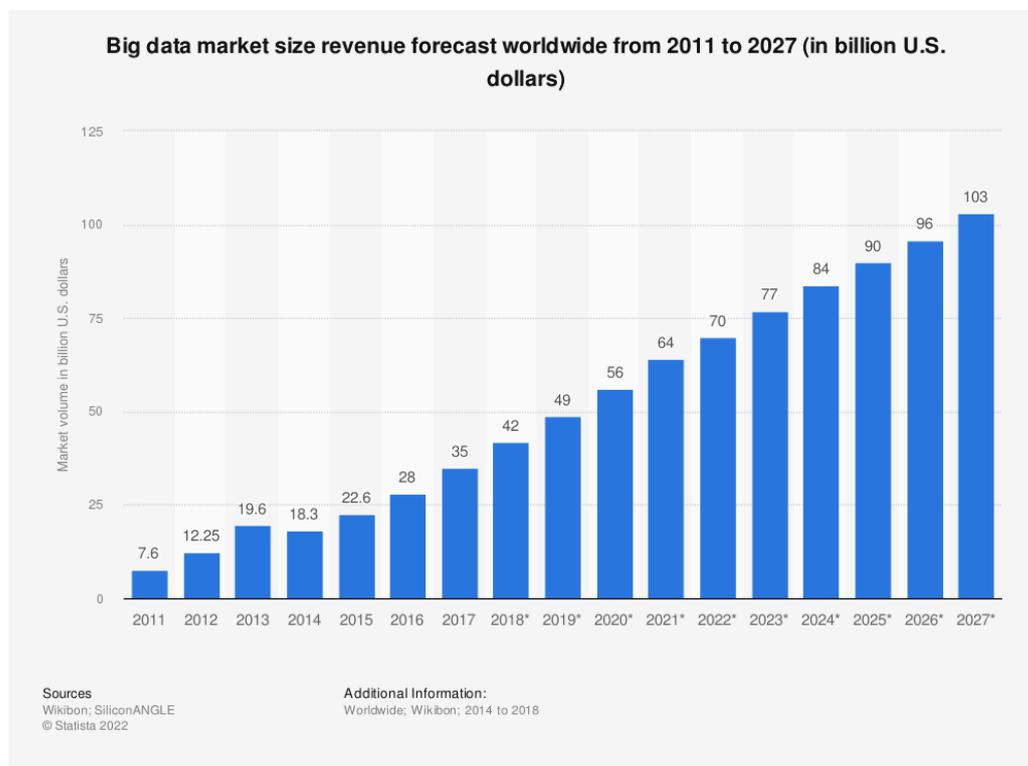


Figura 1.5: Evolución de ingresos del tamaño del mercado de Big Data en todo el mundo[5]

Además, según el informe de CIO España subido el 14 de marzo de 2022, narra que la analítica de datos es un ámbito en constante movimiento y comenta que al principio de la pandemia, parecía que las organizaciones podrían dejar de lado los avances en materia de datos y análisis para centrarse en otras prioridades urgentes, como la habilitación de una fuerza de trabajo remota. Pero, en muchos casos, las organizaciones aceleraron su adopción de capacidades de datos y análisis. En julio de 2020, un estudio de KPMG descubrió que el 67 % de los encuestados aumentó el ritmo de su estrategia de transformación digital debido a la pandemia, y el 63 % aumentó su presupuesto de transformación digital [9].

⁵**Modelo económico:** es una representación simplificada de la relación entre distintas variables que explican cómo opera la economía o un fenómeno en particular de ella.

Por lo que, debido a la gran cantidad de datos que están generando las organizaciones y la inversión que están realizando las empresas en torno al Big Data, se puede considerar que es un buen momento para lanzar al mercado una aplicación con este tipo de prestaciones.

Por otro lado, para lanzar este tipo de aplicación, es necesario conocer la competencia que existe actualmente en el mercado. Por ello, se ha hecho un análisis de las herramientas más famosas que tienen relación con este proyecto:

- **Tableau [10]:** desarrolla productos de visualizaciones de datos interactivos que se enfocan en inteligencia empresarial. A esta se le proporciona los conjuntos de datos que se desea visualizar y crea varias gráficas.
 - **Ventajas:** tiene gran variedad de estilos de gráficas y se puede personalizar fácilmente.
 - **Desventajas:** es una interfaz compleja que requiere tiempo para aprender a utilizarla.
- **Microsoft Power BI [11]:** permite crear visualizaciones de datos de forma individual o dentro de informes cuando se le proporciona uno o varios conjuntos de datos.
 - **Ventajas:** incluye distintos tipos de gráficas y puede analizar datos de diversas fuentes.
 - **Desventajas:** es más adecuado para personas que ya hayan analizado datos anteriormente y hayan trabajado con una interfaz parecida.
- **Google Charts [12]:** es una herramienta basada en la web que puede crear visualizaciones sencillas a partir de conjuntos de datos pequeños y grandes.
 - **Ventajas:** es fácil de usar y se puede personalizar los gráficos con CSS.
 - **Desventajas:** no da un procesamiento estadístico sofisticado.

Todas estas aplicaciones tienen en común que se les proporciona un conjunto de datos y a partir de ello, se genera la gráfica. El problema principal de este tipo de aplicaciones es que no todas las estructuras de los conjuntos de datos son aceptables ya que depende de cómo estén distribuidos los datos dentro del conjunto. Además, parte de ellas requieren cierto conocimiento previo sobre análisis de datos para poder hacer un buen uso de las mismas. También tienen interfaces complejas que llevaría tiempo para aprender a utilizarlas.

Luego de realizar un análisis de las aplicaciones actuales del mercado, se pudo observar que no existe ninguna aplicación enfocada especialmente a conjuntos de datos sobre Canarias, y tampoco existe ninguna aplicación especializada en el empleo, donde se realice directamente el procesamiento y tratamiento de los conjuntos. Esto permitirá diferenciar a este proyecto de las anteriores aplicaciones ya que el usuario podrá ver directamente las visualizaciones y podrá personalizarlas sin necesidad de trabajar con los datos.

1.4. Objetivos

Este estudio pretender responder a las siguientes preguntas: **¿Es viable comercializar una aplicación que visualice ciertos conjuntos de datos sobre el empleo en Canarias? ¿Ayudará realmente a los usuarios que la utilicen?**

Para responder a esas preguntas se han planteado los siguientes objetivos:

- Analizar las herramientas software o tecnologías punteras en el mercado para llevar a cabo el desarrollo del proyecto.
- Seleccionar las tecnologías y entornos de desarrollo definitivos.
- Estudio de los conjuntos de datos relacionados con el empleo en Canarias provenientes de la API de Canarias Datos Abiertos.
- Desarrollo de arquitecturas:
 - Desarrollo de la base de datos.
 - Establecer la base de datos para almacenar los conjuntos de datos procesados.
 - Desarrollo del servidor backend ⁶.
 - Configurar base de datos.
 - Realizar modelos de los conjuntos de datos.
 - Implementar API REST ⁷.
 - Realizar el procesamiento ETL (extraer, transformar y cargar) con los conjuntos de datos obtenidos de la API de Canarias Datos Abiertos.
 - Desarrollo de la aplicación web y móvil.
 - Realizar página principal de dashboard ⁸ que mantenga distintas secciones.
 - Realizar página individual por cada sección donde se muestren todas las gráficas y filtros considerados.
 - Realizar el despliegue del backend y de la aplicación web.
 - Generar el APK ⁹ de la aplicación móvil.
 - Testear el correcto funcionamiento de la versión web y móvil.
- Realizar estudio de la viabilidad económica:

⁶**Backend:** es la parte del desarrollo de software que se encarga del manejo y procesado de los datos y consultas a bases de datos relativas a la aplicación en cuestión. Es la parte de la aplicación que el usuario medio no es consciente de su existencia

⁷**API REST:** es una interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet

⁸**Dashboard:** es una herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los indicadores clave de desempeño (KPI), métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico.

⁹**APK:** corresponden a los términos Android Application Package y son archivos que contienen una aplicación que necesita ser instalada en un dispositivo.

- Crear un Diagrama de Gantt ¹⁰ con las tareas a realizar, su duración y los recursos asignados a cada una.
- Prever la duración del proyecto.
- Prever el costo total del proyecto.
- Considerar nuevas funcionalidades adicionales en un futuro.
- Diseñar un modelo de comercialización del producto.
- Calcular el ROI ¹¹.

¹⁰**Diagrama de Gantt:** es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. No muestra las relaciones entre las diferentes tareas.

¹¹**ROI:** el retorno sobre la inversión (ROI, por las siglas en inglés de return on investment) es una métrica financiera que compara el beneficio o la utilidad obtenida en relación con la inversión realizada, es decir, calcula a partir de qué punto un proyecto comienza a ser rentable recuperando el dinero invertido, tanto al comienzo como durante el proceso.

Capítulo 2

Estudio previo

En el siguiente capítulo se tratarán las diferentes tecnologías o herramientas que se han tomado en cuenta como opciones a la hora de desarrollar el proyecto. Finalmente, se comentará las opciones más adecuadas que se han seleccionado.

2.1. Lenguaje de programación y frameworks

En la actualidad existen varios lenguajes de programación con distintos frameworks ¹ para llevar a cabo el **backend** de la aplicación, pero las tecnologías que se ha tenido en cuenta para desarrollar el servidor son las siguientes:

- **Node.js [13] con el framework Express.js [14]:** está desarrollado principalmente en el lenguaje de JavaScript. Es increíblemente eficiente y liviano. Se suele utilizar para aplicaciones muy intensivas en datos que necesitan operar en tiempo real en equipos distribuidos.
 - **Ventajas:** multiplataforma, ligero, rápido y poseo experiencia en JavaScript.
 - **Desventajas:** la falta de verificación de tipos puede provocar problemas en tiempo de ejecución.
- **Spring Boot [15]:** utiliza el lenguaje Java a través de Spring Boot que permite un rápido inicio de la aplicación y trata de dependencias. Se basa en gran medida en anotaciones. Esto simplifica la configuración cuando un proyecto está creciendo y comienza a tener bastantes dependencias que administrar.
 - **Ventajas:** multiplataforma, se trata de un lenguaje tipado.
 - **Desventajas:** no tengo experiencia en el lenguaje Java.



Figura 2.1: Logos de node.js con Express.js y Spring Boot

¹**Framework:** es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

Para escoger la **base de datos** tuve que tener en cuenta la utilidad que iba a tener en esta aplicación. Podemos destacar dos tipos de bases de datos:

- **Relacionales:** estas organizan la información en partes pequeñas que se integran mediante identificadores.
 - PostgreSQL [16]
- **No Relacionales:** estas no tienen un identificador que sirva para relacionar dos o más partes dentro de una organización.
 - MongoDB [17]



Figura 2.2: Logos de PostgreSQL y MongoDB

Ambos tipos de base de datos son útiles dependiendo de la aplicación en la que se trabaje. Además, he podido trabajar con ambas bases de datos, utilizando el lenguaje SQL para las bases de datos relacionales y manejando objetos JSON para las bases de datos no relacionales. Por lo tanto, en este caso, tuve en cuenta qué tipo de base de datos sería la adecuada en mi aplicación y así poder maximizar su rendimiento.

Por otro lado, para poder escoger el framework que se iba a utilizar para realizar el **frontend de la aplicación web**, se tuvo en cuenta las siguientes tecnologías:

- **ReactJS [18]:** es una biblioteca de Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de single-page application².
 - **Ventajas:** tiene gran nivel de flexibilidad y mayor facilidad para realizar las páginas responsivas y además, es muy ligero ya que los datos del lado del usuario puede representarse en el servidor simultáneamente.
 - **Desventajas:** requiere mucho tiempo para dominarlo y no poseo experiencia previa en este framework.
- **Angular [19]:** es un framework de JavaScript que destaca por la utilización de módulos y también desarrolla aplicaciones de página única.
 - **Ventajas:** tiene ciertas inyecciones de dependencias de las características relacionadas a los componentes de los módulos y la modularidad en general y además, poseo experiencia en este framework.
 - **Desventajas:** tiene una sintaxis compleja.

²**single-page application:** sus siglas son SPA y es también conocido como aplicación de página única. Consiste en una aplicación web que cabe en una sola página con el propósito de dar una experiencia más fluida a los usuarios.

- **VueJS [20]:** es un framework de JavaScript que encaja perfectamente en la creación de aplicaciones con interfaces de usuario altamente adaptables y también desarrolla aplicaciones en una sola página.
 - **Ventajas:** tiene una excelente integración, pequeño tamaño y se utiliza bastante HTML. Además, poseo un poco de experiencia en este framework.
 - **Desventajas:** puede tener problemas para integrarse en grandes proyectos debido a su alta flexibilidad.



Figura 2.3: Logos de ReactJS, Angular y VueJS

Por otro lado, las opciones que se tuvieron en cuenta para realizar el **frontend de la aplicación móvil** en Android son las siguientes:

- **React Native [21]:** es un framework de código abierto que combina el desarrollo nativo con React y se utiliza para desarrollar aplicaciones móviles.
 - **Ventajas:** permite crear un único código para ambas versiones en vez de crear dos aplicaciones diferentes para Android y iOS y además, el marco está escrito en JavaScript.
 - **Desventajas:** tiene alta complejidad, dificultad de compatibilidad y depuración y además, no tengo experiencia en este framework.
- **Ionic [22]:** es un SDK ³ de frontend de código abierto que permite desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones para iOS nativo, Android y web, desde una única base de código.
 - **Ventajas:** se integra con los principales frameworks de frontend, como Angular, React y Vue y ofrece un diseño limpio, sencillo y funcional.
 - **Desventajas:** no poseo experiencia en este framework.

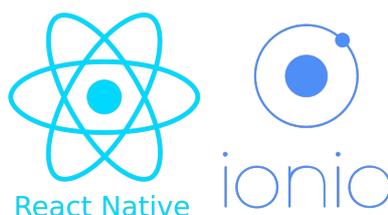


Figura 2.4: Logos de React Native e Ionic

³**SDK:** por sus siglas es kit de desarrollo de software (SDK), que trata de un conjunto de herramientas proporcionado usualmente por el fabricante de una plataforma de hardware, un sistema operativo (SO) o un lenguaje de programación.

Finalmente, a la hora de decidir qué **librería para visualizar los datos** podía encajar mejor, se tuvo en cuenta las siguientes:

- **D3.js [23]**: es una librería de gráficos de JavaScript que permite a los desarrolladores crear visualizaciones de datos interactivas y personalizadas.
 - **Ventajas**: se pueden personalizar los gráficos o incluso crearlos desde cero.
 - **Desventajas**: tiene alta complejidad y no poseo experiencia con esta tecnología.
- **Chart.js [24]**: proporciona gráficos de barras, de líneas, circulares y de dispersión. Además, representa sus gráficos utilizando el elemento Canvas⁴, lo que da como resultado un buen rendimiento.
 - **Ventajas**: se requiere una cantidad mínima de codificación y poseo experiencia con esta librería. Se integra bien con React utilizando una librería llamada react-chartjs-2 [25].
 - **Desventajas**: no es posible personalizar desde cero un gráfico.



Figura 2.5: Logos de D3.js y Chart.js

2.2. Entorno de desarrollo

En cuanto al entorno de desarrollo no hubo distintas opciones comparativas y se utilizaron las siguientes:

- **Plataforma API de Canarias Datos Abiertos**: se trata de una plataforma que contiene todos los datos del sector público canario. Se ha utilizado esta API para extraer los datos ya que en el mercado, no existe ninguna otra API especializada en los conjuntos de datos a nivel de Canarias, y debido a que nos interesan sólo los datos de las islas, se ha optado directamente por hacer uso de esta. Existen otras fuentes a nivel nacional como la API de datos del Gobierno de España [26]. Esta última también contiene algunos conjuntos de datos de las islas pero la propia API de Canarias contiene mayor información.
- **GitHub [27]**: se trata de repositorios donde se alojan proyectos utilizando el sistema de control de versiones Git. Se ha utilizado para alojar los códigos de las aplicaciones.
- **Visual Studio Code [28]**: es un editor de código fuente desarrollado por Microsoft. Es de software libre y multiplataforma, tiene una buena integración con Git y cuenta con soporte para depuración de código.

⁴**Canvas**: es un elemento HTML incorporado en HTML5 que permite la generación de gráficos permitiendo la renderización dinámica de gráficos 2D y mapas de bits, así como animaciones con estos gráficos.

- **Android Studio [29]:** es el entorno de desarrollo integrado oficial para la plataforma Android. Fue creado específicamente para crear aplicaciones en Android. Contiene un emulador para probar la aplicación en distintos dispositivos.



Figura 2.6: Logos de GitHub, Visual Studio Code y Android Studio

- **Heroku [30]:** es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación. Permite manejar los servidores y sus configuraciones, escalamiento y la administración. Esta plataforma ya la había utilizado anteriormente para realizar los despliegues y tiene un buen correcto funcionamiento.
- **Thunder Client [31]:** es una extensión de Visual Studio Code que permite simular llamadas a las API.



Figura 2.7: Logos de Heroku y Thunder Client

Cabe destacar que para utilizar una plataforma donde administrar la base de datos de manera sencilla, se tuvo en cuenta los siguientes dos entornos dependiendo de la base de datos seleccionada:

- **MongoDB Atlas [32]:** es un servicio de Cloud Database (o Base de Datos en la Nube), que te permite crear y administrar la base de datos de Mongo desde cualquier lugar del mundo, a través de su plataforma.
- **pgAdmin [33]:** es la plataforma de desarrollo y administración en funciones para PostgreSQL.



Figura 2.8: Logos de MongoDB Atlas y pgAdmin

2.3. Elección de tecnologías y entorno de desarrollo

Finalmente, para llevar a cabo el desarrollo del proyecto se ha decidido utilizar las siguientes herramientas:

- **Backend:** se ha utilizado Node.js con Express.js debido a la experiencia que he tenido en este tipo de backend y además, me proporciona eficiencia y comodidad.
- **Base de datos:** en este caso, debido a que la información que se almacena en la base de datos no tiene relación entre sí, se ha utilizado la base de datos no relacional de MongoDB y por lo tanto, el entorno de desarrollo para administrar la base de datos es MongoDB Atlas.
- **Frontend web:** se ha utilizado ReactJS dado que no tengo ningún tipo de experiencia en este framework y me gustaría aprender acerca de esta tecnología. Además, según la gráfica siguiente originada por Stack Overflow [34] se puede observar que este framework está siendo el más pronunciado entre la comunidad.

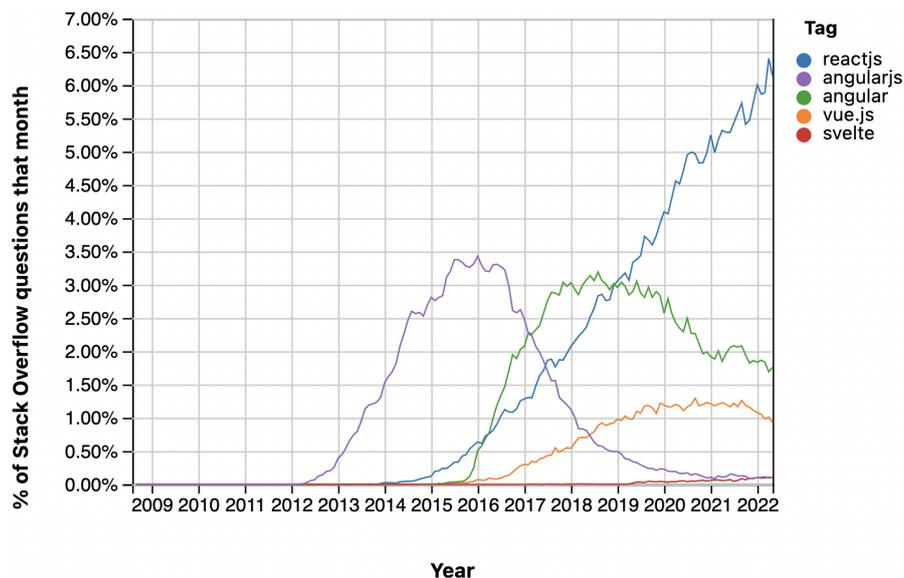


Figura 2.9: Porcentaje de dudas de frameworks más preguntados durante los años [6]

- **Frontend móvil:** se ha utilizado Ionic basado en la integración de React ya que me permite trabajar con el mismo framework que en el frontend web y es una aplicación que permite la escalabilidad fácilmente.
- **Librería para visualizar datos:** se ha utilizado ChartJS dado que se encontraba entre los más utilizados dentro del mercado por su facilidad para integrar los gráficos. Además, no es necesario personalizar los gráficos en profundidad para representar los datos o incluso crear otros nuevos, por lo que, con los gráficos que ofrecía se podía representar perfectamente la información. Cabe destacar que para mejorar y personalizar las leyendas se ha tenido en cuenta el uso de un plugin de esta librería, llamado chartjs-plugin-datalabels [35].

2.4. Dificultades encontradas en las tecnologías

En cuanto a las tecnologías utilizadas para el frontend de la aplicación móvil tuve cierta dificultad en esta elección. En principio comencé a utilizar React Native pero debido a la alta complejidad que tenía y la poca adaptación de los componentes de la aplicación web de ReactJS, prefería utilizar Ionic ya que me permitía adaptar de forma más flexible los componentes realizados en el frontend de la aplicación web.

Por otro lado, a cerca de la librería de visualizaciones de datos, en principio utilicé la librería D3.js pero debido a la alta complejidad para realizar un gráfico de barras simple, no vi necesario utilizar esta herramienta sabiendo que ChartJS tenía buenas prestaciones y menor dificultad para realizar gráficos sencillos. Además, la librería ChartJS mantiene una extensión que permite integrar los gráficos como componentes en el framework de ReactJS.

Capítulo 3

Desarrollo de arquitecturas

En este apartado se explicará el procesamiento y tratamiento de los datos así como la estructura de la aplicación web y aplicación móvil.

3.1. Procesamiento de datos

En primer lugar, se va a tratar sobre el procesamiento de datos que se ha llevado a cabo sobre los conjuntos de datos de la API de Canarias Datos Abiertos.

3.1.1. Extracción, transformación y carga (ETL)

Dentro del primer proceso de ETL ¹, tenemos la extracción de la fuente de datos de Canarias Datos Abiertos. Esta API corresponde a un punto de encuentro para la publicación de datos del sector público canario en formatos abiertos, gratuitos y reutilizables.

Para poder realizar las llamadas necesarias a la API y que devuelva el conjunto de datos deseado en formato JSON, se tuvo que estudiar la documentación de la API [36]. En primer lugar, para poder acceder al catálogo general de datos, se ha hecho uso del Thunder Client que permite realizar llamadas a la API. El punto de entrada para acceder al catálogo general es el siguiente:

`https://datos.canarias.es/catalogos/general/api/action`

Para poder llevar a cabo la extracción de un conjunto de datos en específico es necesario obtener el id del paquete. Para ello, se accede al recurso que se desea extraer en la página web de Canarias Datos Abiertos. Existen distintas distribuciones disponibles pero la necesaria es la distribución JSON para poder manipular los datos y luego almacenarlos en la base de datos de MongoDB.

Al acceder sobre la distribución JSON del conjunto de datos, se muestra una tabla de información adicional que proporciona el id temporal. Este no corresponde al id del paquete ya que el id temporal cambia cuando el conjunto de datos es actualizado, mientras que el id del paquete es fijo desde su origen. Para obtener el id del paquete, es necesario

¹**ETL**: sus siglas significan en inglés Extract, Transform and Load; Son un conjunto de tecnologías, herramientas y técnicas que permiten extraer grandes volúmenes de datos de múltiples fuentes, incluyendo su transformación y carga en un almacén final.

añadir al punto de entrada nombrado anteriormente el comando `resource_show` y el parámetro del id temporal. Al realizar la llamada, devuelve un formato JSON que contiene la información del conjunto de datos donde se puede extraer el id de origen del paquete.

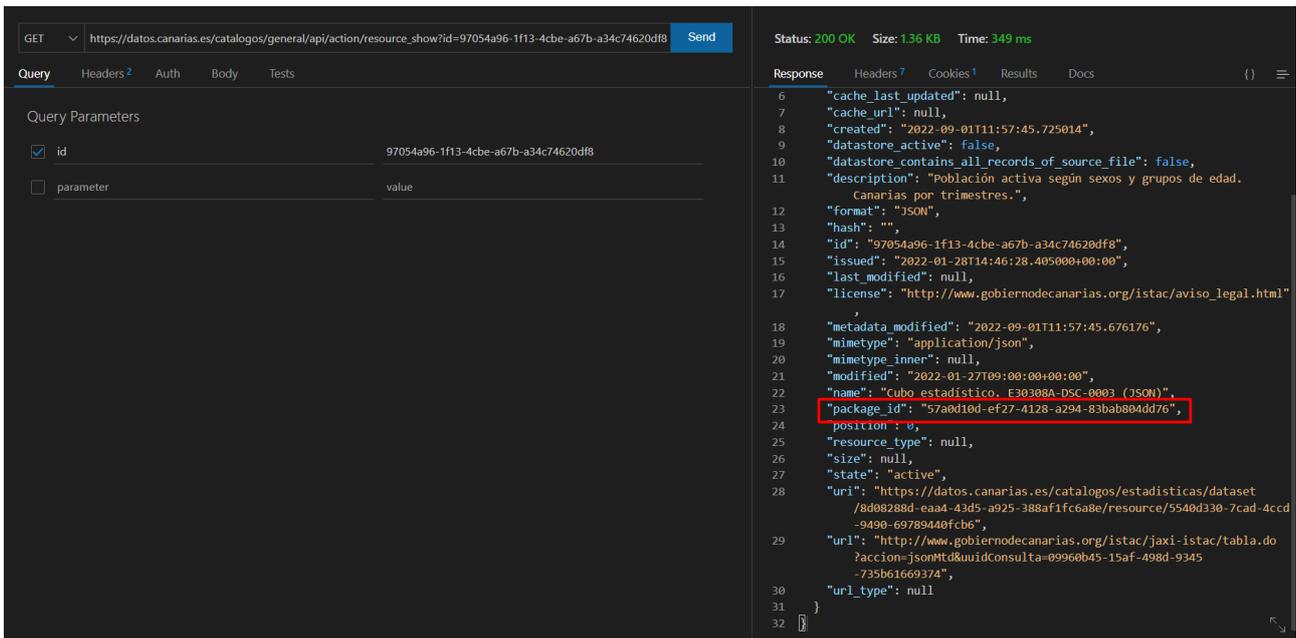


Figura 3.1: Uso del Thunder Client para obtener el id de origen del paquete

Una vez conocido el id del paquete, con el comando `package_show` se obtiene la informaci\u00f3n de los distintos formatos del paquete y finalmente, se puede adquirir el acceso a la url que devuelve el contenido de datos del conjunto en formato JSON.

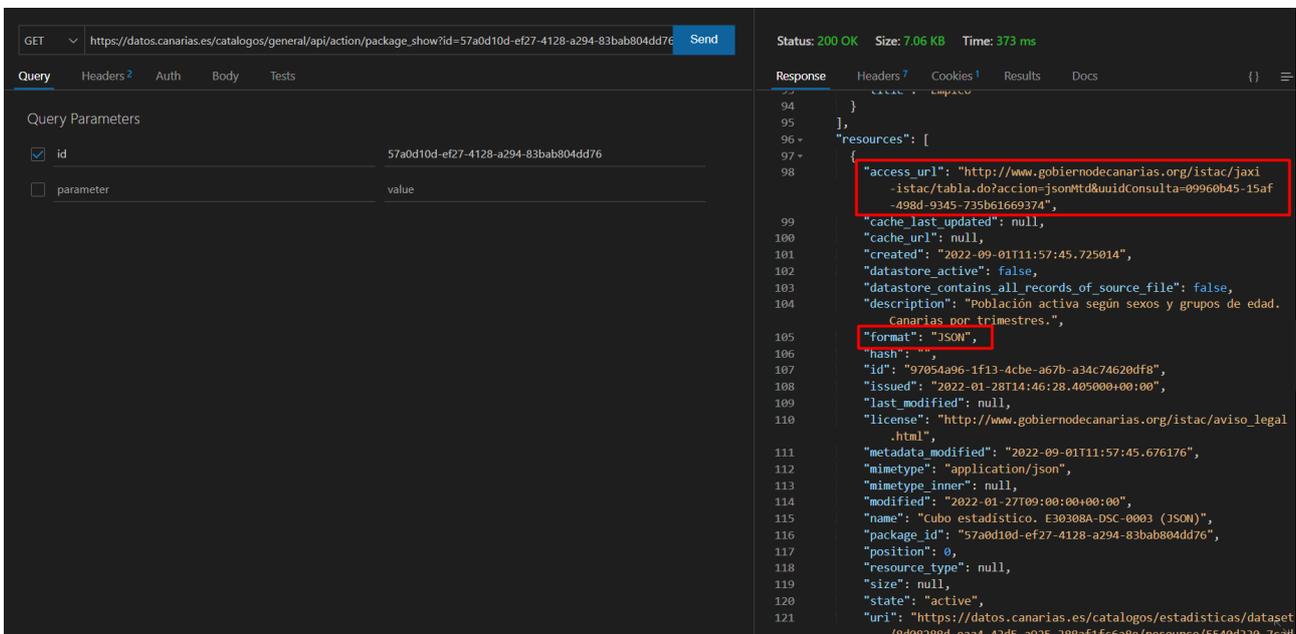


Figura 3.2: Uso del Thunder Client para obtener la url que devuelve el contenido de datos

A la hora de implementar las llamadas a la API en el servidor backend, no har\u00e1 falta entrar en la p\u00e1gina web de Canarias Datos Abiertos y obtener el id temporal del conjunto de datos, ya que al conocer el id del paquete de origen que se mantiene siempre

igual, se podrá extraer el conjunto a partir del comando *package_show* como se mostró anteriormente.

El siguiente proceso ETL es la transformación de los datos. Una vez que en el backend se obtiene el conjunto de datos, este se procesa dependiendo de la estructura de datos que requiera la librería de visualizaciones de datos que se explicará más adelante.

Finalmente, en el último proceso ETL se realiza la carga, es decir, que una vez se procesan los datos, se almacenan en la base de datos de MongoDB.

3.1.2. Creación de nuevos conjuntos de datos

Además del procesamiento de los conjuntos de datos extraídos de la API, también se lleva a cabo el tratamiento de nueva información proveniente de la combinación de dos conjuntos.

Por lo que, se obtiene un nuevo conjunto cuando existen dos conjuntos de datos de la API que tienen relación y de los cuales, mediante ciertos cálculos aritméticos, se puede obtener información nueva.

De esta forma, este proyecto de aplicación, además de extraer, procesar, cargar y visualizar conjuntos de datos de la API de Canarias Datos Abiertos, también creará nuevos conjuntos a partir de los ya existentes, obteniendo nueva información que pueda ser visualizada. De esta manera, se ofrecerá al usuario un mayor conocimiento.

3.1.3. Actualización de conjuntos de datos

Para que la aplicación esté constantemente actualizada, se almacena en la base de datos el último periodo del que se ha obtenido información sobre el conjunto de datos, es decir, que se almacena el último periodo del que se obtiene valor dentro del conjunto.

Luego, cada cierto tiempo, se lanza una llamada que comprueba si el conjunto de datos de la API se ha actualizado con nuevos valores ya que necesitamos mantener actualizado el conjunto almacenado en la base de datos. Para ello, se compara el último periodo almacenado en la base de datos, y el último periodo que se obtiene al hacer la llamada a la API sobre ese conjunto. En caso de que ambos periodos no sean iguales, se debe de actualizar el conjunto de datos almacenado en la base de datos.

Para comprobar que los conjuntos están actualizados, se programa una función que se ejecuta todos los lunes a las 00:00 h. En caso de que algún conjunto de datos esté desactualizado, se actualizará con los datos de la API de Canarias Datos Abiertos.

Los nuevos conjuntos de datos que son creados a partir de otros conjuntos de datos ya almacenados, se actualizarán también en caso de que los dos conjuntos padre de los que proviene hayan sido actualizados. Es necesario que ambos estén actualizados ya que se debe realizar la operación aritmética sobre ellos para obtener los nuevos valores. Por ejemplo, si los conjuntos de datos padre tienen como último periodo almacenado 2021 y 2022 respectivamente, no se podrá obtener el nuevo valor del periodo 2022 ya que el primer conjunto padre no tiene un valor almacenado sobre ese periodo y por lo tanto, no se puede realizar el cálculo para obtener el nuevo valor. Cabe destacar que estos se

actualizarán 15 minutos más tarde, es decir, a las 00:15 h ya que de esta forma, se evitan problemas de concurrencia² y se actualizan cuando los dos conjuntos de datos de los que proviene se hayan actualizado en la base de datos.

3.2. Estructura backend

El backend utilizado en el proyecto está realizado en Node.js con el framework Express.js. Con este framework se ha podido implementar el diseño de API REST simplificando el control de las rutas y peticiones.

El concepto de API REST es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP, esta permite el intercambio de la información entre el backend y el frontend de la aplicación.

Técnicamente, la aplicación del servidor y del cliente son independientes y la única información que la aplicación de cliente conoce es la URI o también conocido endpoint del recurso solicitado; no puede interactuar con la aplicación de servidor de ninguna otra manera. De forma similar, una aplicación de servidor no debe modificar la aplicación de cliente aparte de pasarle los datos solicitados a través de HTTP.

Al tratarse de manera independiente el servidor backend, el código fuente de este se encuentra almacenado en un repositorio de GitHub³.

La estructura del backend se basa en el diseño API REST y contiene las siguientes carpetas y ficheros:

- **models:** contempla esquemas, es decir, la estructura de datos que se almacena por cada conjunto de datos. Esta estructura de datos está basada en distintos arrays.
- **node_modules:** contiene las dependencias del proyecto.
- **src:** se trata del código fuente del proyecto.
 - **routes:** gestiona los nombres de los puntos de entrada o endpoints por cada conjunto de datos. Esto es necesario ya que, a través de esta URI, el backend realiza la acción de traer, añadir, modificar o eliminar un modelo almacenado en la base de datos. Esta carpeta se relaciona con los controllers que ejercen una de las acciones anteriores, una vez que se hace una llamada a ese punto de entrada desde el cliente.
 - **controllers:** contiene el código necesario para responder a las acciones que se solicitan en la aplicación. En este caso, para cada conjunto de datos existe una función HTTP GET que permite encontrar el modelo o conjunto de datos que se desea buscar en la base de datos y devuelve dicho modelo en un objeto JSON.

²**Concurrencia:** la programación concurrente no es más que la forma en la cual podemos resolver ciertas problemáticas de forma concurrente, es decir, ejecutando múltiples tareas a la misma vez y no de forma secuencial.

³**Repositorio Backend en GitHub:** <https://github.com/alu0101228020/visualizaBackend.git>

- **process:** en esta carpeta se lleva a cabo el procesamiento de los conjuntos de datos. En primer lugar, estos ficheros contienen la función del node-cron [7] que trata de un pequeño programador de tareas en JavaScript puro para Node.js. Este se encarga de ejecutar la función que comprueba si los conjuntos de datos de la base de datos están actualizados respecto a los de la API y en caso de que estén desactualizados, realizan el procesamiento de los datos para actualizarlos en la base de datos.

Para lanzar la función del node-cron, se debe especificar el periodo en el que se quiere ejecutar. La estructura que debe tener es la siguiente:

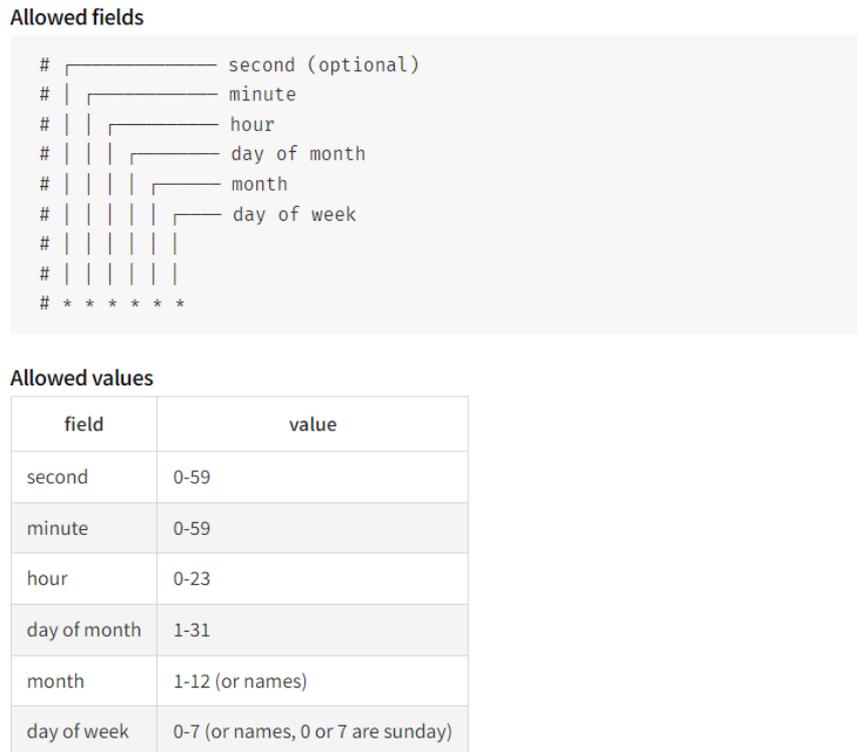


Figura 3.3: Estructura de cadena para representar el periodo de node-cron [7]

Como se dijo anteriormente, se desea que se lance la función cada lunes a las 00:00 h. Entonces la estructura correspondería a la siguiente: "0 0 0 * 1", siendo el uno el día lunes y los ceros la hora. Mientras que para los nuevos conjuntos de datos creados a partir de los ya existentes, se tendría que especificar el node-cron de la manera "0 15 0 * * 1", que se traduciría a las 00:15 h de cada lunes.

Ahora bien, debido a que existen dos tipos de procesamiento dependiendo de si se trata de un conjunto de la API o de un nuevo conjunto, se explicará el tratamiento de los datos por separado:

Procesamiento de un conjunto de datos de la API

En primer lugar, se extrae el conjunto de datos a través de la llamada a la API como se explicó anteriormente y por otro lado, se extrae el conjunto de datos

almacenado en la base de datos. Para realizar estas llamadas se utiliza Axios [37] que es un Cliente HTTP basado en promesas para Node.js. Una vez se obtienen ambos conjuntos, se comprueba si el conjunto de la base de datos está actualizado respecto al de la API. Para ello, se compara el campo almacenado en la base de datos que contiene el último periodo que tiene valor, con el último periodo que se obtiene del conjunto de la API.

En caso de que el conjunto de datos necesite ser actualizado, se lleva a cabo el procesamiento de los datos a través de la llamada a la función **processDataset** que se encuentra en el mismo fichero. A esta función se le pasa el conjunto que se quiere procesar. Dentro de esta función se procesan los datos filtrando los valores y almacenándolos en distintos arrays. Una vez procesado, se llama al servicio que almacena la estructura JSON en la base de datos.

Procesamiento de un nuevo conjunto de datos

En este caso, se realizan dos llamadas a la base de datos para obtener los conjuntos de los que proviene este nuevo conjunto. En caso de que ambos conjuntos padre hayan sido actualizados, es decir, que los valores del último periodo almacenado en la base de datos sea distinto al del nuevo conjunto, este último se actualiza. Es necesario que ambos conjuntos padre estén actualizados ya que si alguno de los conjuntos tiene un periodo almacenado distinto al otro, no se podrá realizar la operación aritmética que devuelva el nuevo conjunto actualizado.

En caso de que ambos conjuntos padre estén actualizados y por lo tanto, tengan el mismo último periodo almacenado en la base de datos, y este no sea igual al último periodo del nuevo conjunto, este último se actualiza. Para ello, primero se llama a la función **processDataset** que filtra los datos de ambos conjuntos padre y realiza la operación aritmética necesaria para obtener el nuevo conjunto. Estos son almacenados en estructuras de arrays. Finalmente, se llama al servicio que almacena la estructura JSON en la base de datos.

- **services:** son los servicios que administran la base de datos. Por cada conjunto de datos existe una función llamada **insertData** que al pasarle por parámetro un objeto JSON, lo asocia al modelo de un conjunto de datos y lo guarda en la colección de la base de datos de ese conjunto. Estos servicios son llamados, una vez que se procesan los datos y se almacenan en una estructura JSON como se comentó anteriormente.
- **utils:** almacena las constantes del node-cron para establecer la fecha de la ejecución de las tareas.
- **database.js:** contiene la configuración de la base de datos de MongoDB.
- **index.js:** fichero principal donde se declaran los archivos y donde se abre el puerto de escucha para las peticiones del servidor.
- **.gitignore:** permite ignorar aquellas carpetas o archivos que son pesados para subirlos a GitHub y pueden ser generados, como el *node_modules*.
- **package-lock.json** y **package.json:** ficheros de configuración de Node.js.

3.3. Estructura librería de visualización de datos

La librería de visualizaciones de datos utilizada para el proyecto es ChartJS específicamente react-chartjs-2 ya que esta última se trata de una extensión de ChartJS adaptada al framework de ReactJS.

Debido a que ReactJS es un framework de componentes, a través de la librería se importan los tipos de gráficos que se desean utilizar. A cada componente se le incorpora el *data* que corresponde a un objeto JSON. Este objeto JSON tiene un campo *labels* que es un array de etiquetas en formato cadena y por otro lado, tiene un campo llamado *datasets* que trata de un array de objetos que representarían los datos, donde cada objeto debe tener uno o varios arrays numéricos dependiendo del tipo de gráfico que se quiera implementar.

Por lo que, la estructura necesaria para visualizar los datos en esta librería trata de arrays de tipo numérico para los datos y arrays de tipo cadena para las etiquetas.

También, se le añade el campo de *options* que trata de un objeto JSON que contiene las opciones del gráfico para llevar a cabo la personalización. Finalmente, si se desea personalizar las leyendas del gráfico, se utilizaría las opciones ofrecidas por la librería *chartjs-plugin-datalabels* que se añade como plugin al componente.

```
<Pie data={dataPoblacionParada} plugins={[ChartDataLabels]} options={optionsPoblacionParada}/>
```

Figura 3.4: Componente Pie de la librería react-chartjs-2

Los tipos de gráficos importados y utilizados de la librería son los siguientes:

- **Bar Chart:** trata de un gráfico de barras que proporciona una forma de mostrar valores de datos representados como barras verticales. A veces se usa para mostrar datos de tendencias y la comparación de múltiples conjuntos de datos uno al lado del otro [38].

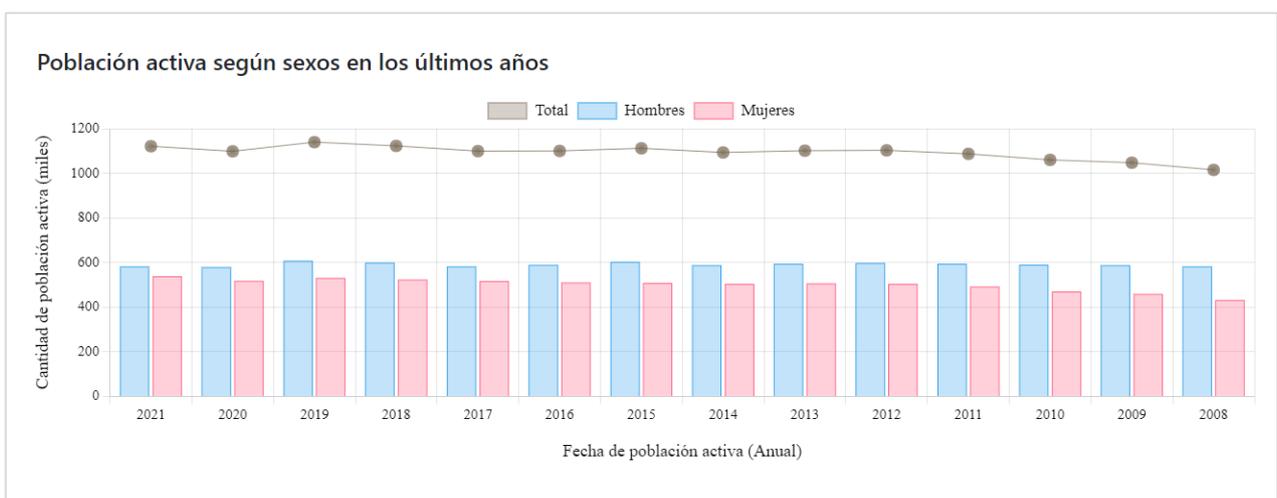


Figura 3.5: Gráfica de la aplicación de tipo Bar Chart

- **Doughnut and Pie Charts:** es un gráfico dividido en segmentos, el arco de cada segmento muestra el valor proporcional de cada dato. Son excelentes para mostrar las proporciones relacionales entre los datos [39].

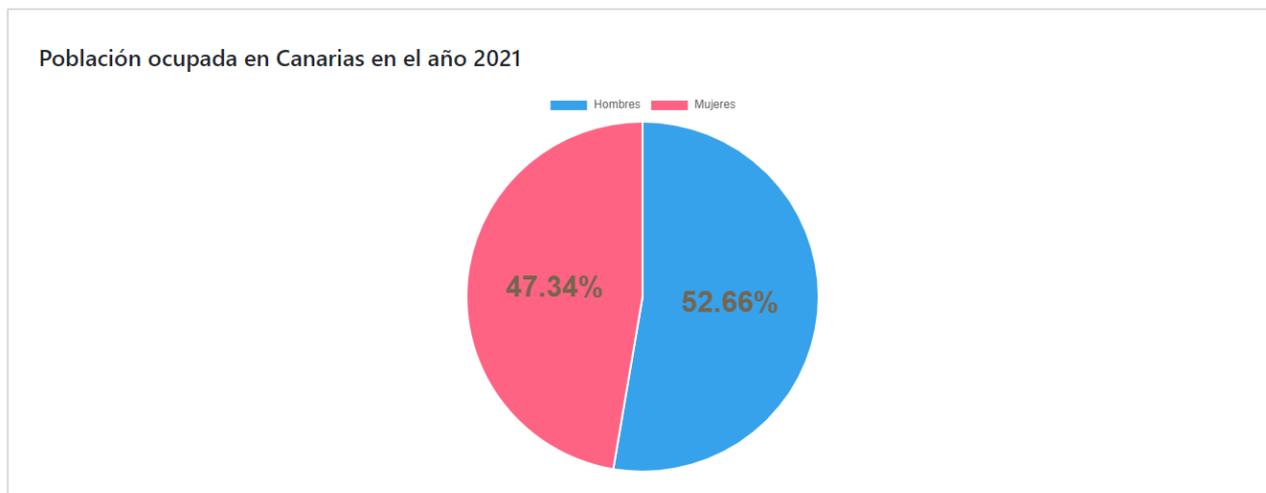


Figura 3.6: Gráfica de la aplicación de tipo Pie Chart

- **Line Chart:** trata de un gráfico de líneas. A menudo, se utiliza para mostrar datos de tendencias o la comparación de dos conjuntos de datos [40].

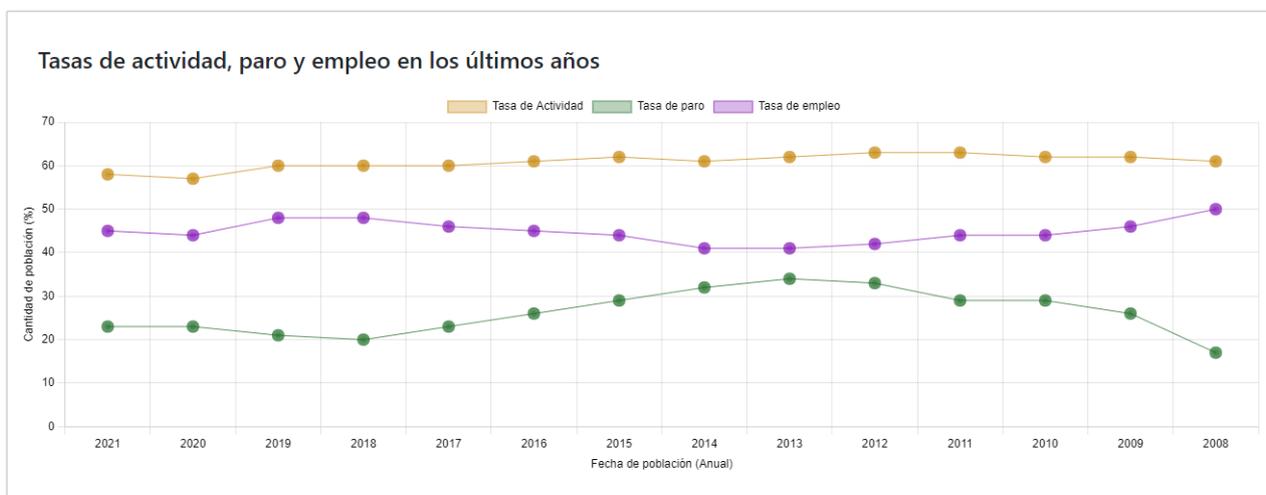


Figura 3.7: Gráfica de la aplicación de tipo Line Chart

- **Polar Area Chart:** estos son similares a los gráficos circulares, pero cada segmento tiene el mismo ángulo: el radio del segmento difiere según el valor. Este tipo de gráfico suele ser útil cuando queremos mostrar una comparación de datos similar a un gráfico circular, pero también puede mostrar una escala de valores para el contexto [41].

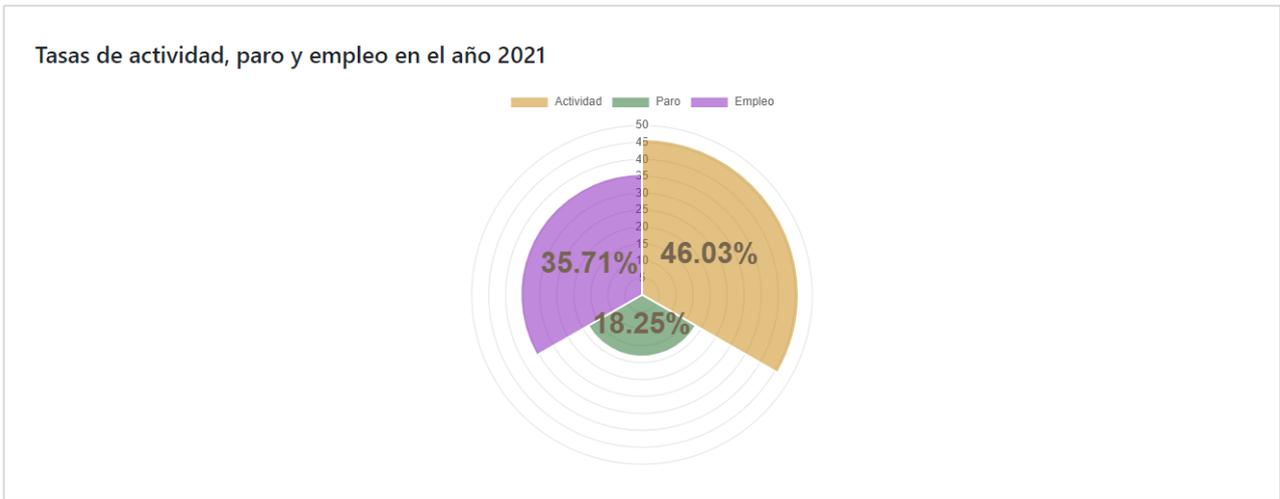


Figura 3.8: Gráfica de la aplicación de tipo Polar Area Chart

- Radar Chart:** un gráfico de radar es una forma de mostrar múltiples puntos de datos y la variación entre ellos. A menudo son útiles para comparar los puntos de dos o más conjuntos de datos diferentes [42].

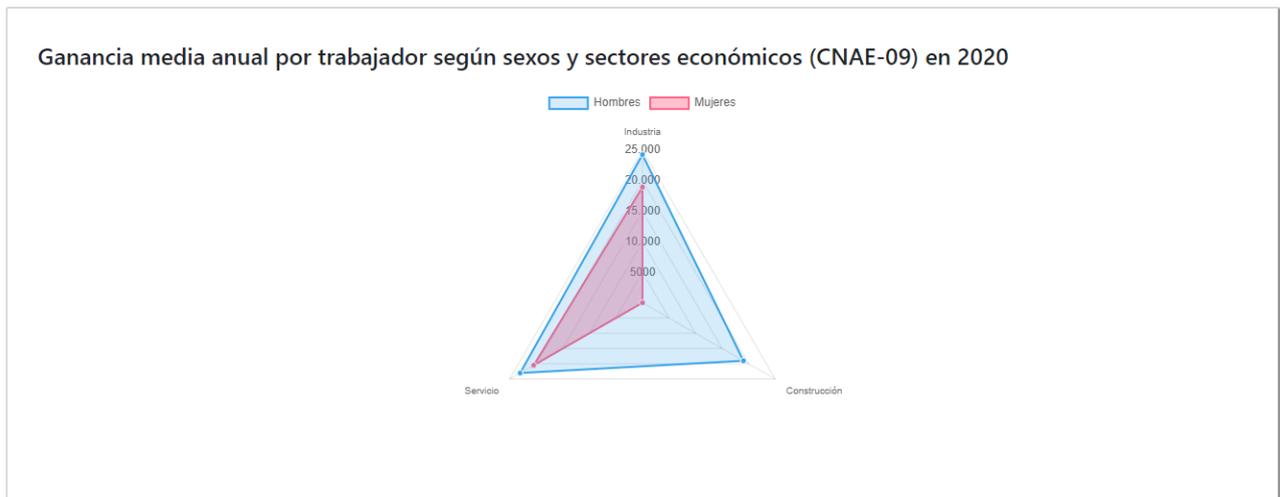


Figura 3.9: Gráfica de la aplicación de tipo Radar Chart

- Mixed Chart Types:** permite crear gráficos mixtos que son una combinación de dos o más tipos de gráficos diferentes [43].

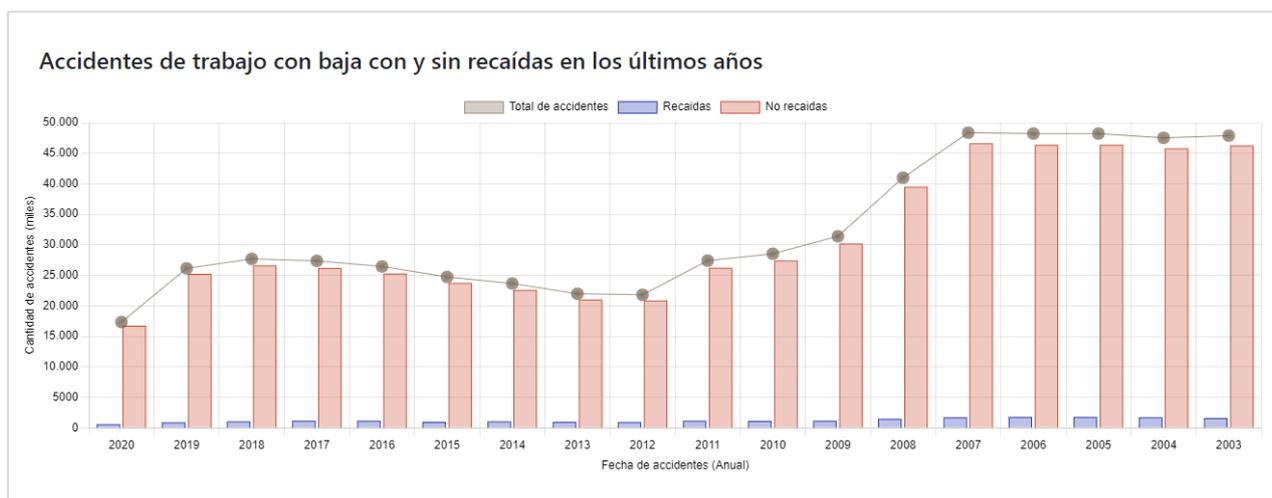


Figura 3.10: Gráfica de la aplicación que combina los tipos Bar Chart y Line Chart

3.4. Estructura base de datos

La base de datos utilizada es MongoDB Atlas cuyo servicio de Base de Datos en la Nube permite crear y administrar la base de datos de manera sencilla a través de la plataforma.

Cada colección almacenada en la base de datos corresponde a los siguientes conjuntos de datos:

- **Conjuntos de datos extraídos y procesados de la API de Canarias Datos Abiertos:**

- Accidentes de trabajo con baja según gravedad y sexos. Comarcas de Canarias y años.
- Tasas de actividad, paro y empleo según sexos y grupos de edad. Canarias por trimestres.
- Ganancia media anual por trabajador según sexos y sectores económicos (CNAE-09). España y Canarias por años.
- Población activa según sexos y grupos de edad. Canarias por trimestres.
- Población parada según sexos y grupos de edad. Canarias por trimestres.
- Recaídas de accidentes de trabajo con baja según gravedad y sexos. Comarcas de Canarias y años.

- **Nuevos conjuntos de datos generados a partir de la relación de dos conjuntos de datos existentes:**

- No recaídas de accidentes de trabajo con baja según gravedad y sexos. Comarcas de Canarias y años. Este fue generado a partir de la resta del conjunto de datos de accidentes de trabajo con baja según gravedad y sexos, y del conjunto de recaídas de accidentes de trabajo con baja según gravedad y sexos.
- Población ocupada según sexos y grupos de edad. Canarias por trimestres. Este fue generado a partir de la resta del conjunto de datos de población activa según sexos y grupos de edad y del conjunto de población parada según sexos y grupos de edad.

Si se desea ver los recursos de los distintos conjuntos de datos de la API de Canarias Datos Abiertos, se pueden encontrar en el fichero *README.md* del servidor backend en el repositorio de GitHub.

La estructura que se ha seguido para almacenar la información en la base de datos corresponde a la utilizada por la librería de visualizaciones de datos, ChartJS. Cada array de este conjunto, almacena la información del mismo bajo uno o varios determinados filtros y a su vez, dentro de cada array se almacena en la primera posición un array numérico con los datos y en la segunda posición, un array de cadena con las etiquetas.

```

  _id: ObjectId("62e02ba76c2a8578ab3a9936")
  annualTotalTotal: Array
    0: Array
      0: 17345
      1: 26141
      2: 27698
      3: 27377
      4: 26465
      5: 24740
      6: 23669
      7: 21985
      8: 21815
      9: 27408
      10: 28544
      11: 31407
      12: 41012
      13: 48373
      14: 48241
      15: 48238
      16: 47542
      17: 47899
    1: Array
      0: "2020"
      1: "2019"
      2: "2018"
      3: "2017"

```

Figura 3.11: Ejemplo de conjunto de datos almacenado (1)

Además, cada conjunto de datos almacena el último periodo del que se ha obtenido datos en el campo **dateModified**. Este campo es el que se utiliza para comparar si el conjunto almacenado está actualizado con respecto al conjunto almacenado en la API.

```

  _id: ObjectId("631627559b0972942cb8c448")
  > annualTotalTotal: Array
  > annualMenTotal: Array
  > annualWomenTotal: Array
  > annualTotalMild: Array
  > annualMenMild: Array
  > annualWomenMild: Array
  > annualTotalSerious: Array
  > annualMenSerious: Array
  > annualWomenSerious: Array
  > annualTotalMortal: Array
  > annualMenMortal: Array
  > annualWomenMortal: Array
  dateModified: "2020"
  __v: 0

```

Figura 3.12: Ejemplo de conjunto de datos almacenado (2)

3.5. Estructura frontend

En las siguientes secciones se puede observar el frontend de la aplicación web y móvil.

3.5.1. Frontend web

El desarrollo del frontend web de la aplicación se ha llevado a cabo utilizando principalmente React, CSS y JavaScript. El código fuente se ha almacenado en un repositorio en GitHub⁴. A continuación, se destacará ciertas partes de la estructura de carpetas del frontend web:

- **public:** contiene aquellas imágenes e iconos que componen la aplicación.
- **src:** se trata del código fuente de la aplicación.
 - **components:** contiene los componentes de la aplicación. Existe un archivo js y css por cada sección existente.
- **pages:** contiene las páginas de la aplicación y también se almacena un archivo js y css por cada página.
- **App.js y App.css:** corresponde a la página principal donde se almacenan las rutas de las secciones.

La aplicación de este proyecto se llama **Visualiza** y en la web se tienen las siguientes páginas:

Dashboard

La página principal corresponde al dashboard de la aplicación.

En primer lugar, se observa un diseño sencillo e intuitivo. De primeras, se dispone de una previsualización de las distintas secciones que componen la aplicación: Poblaciones, Tasas, Ganancias y Bajas Laborales. Esto permite al usuario tener una visión general de los distintos temas existentes y además, permite tener estructura organizativa de la información.

Para acceder a una sección en específico, se debe pulsar clic sobre la sección. Además, existe un foco de sombra que permite saber sobre qué sección se tiene el foco del ratón.

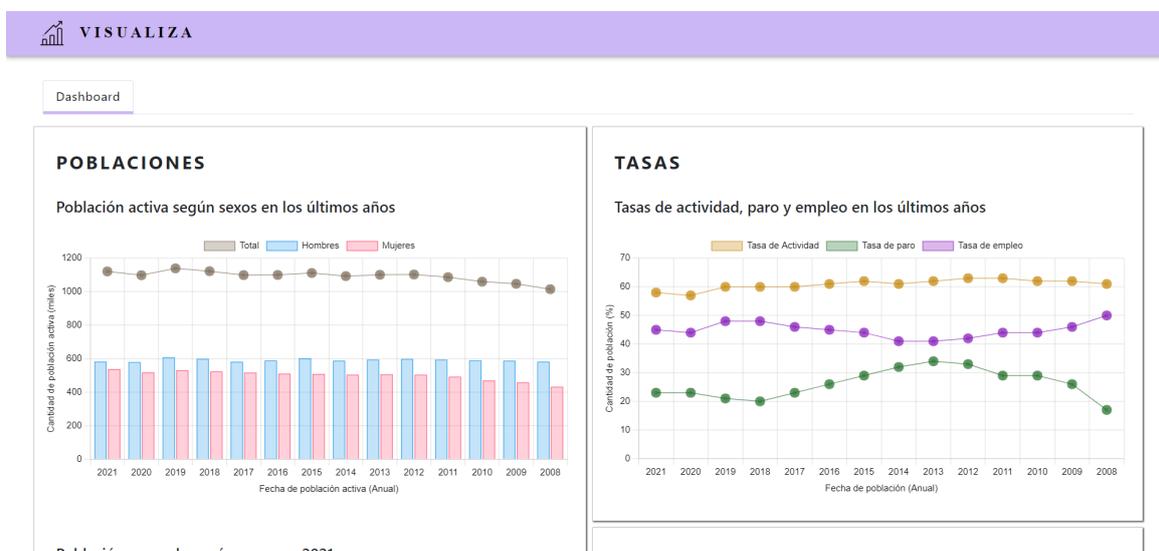
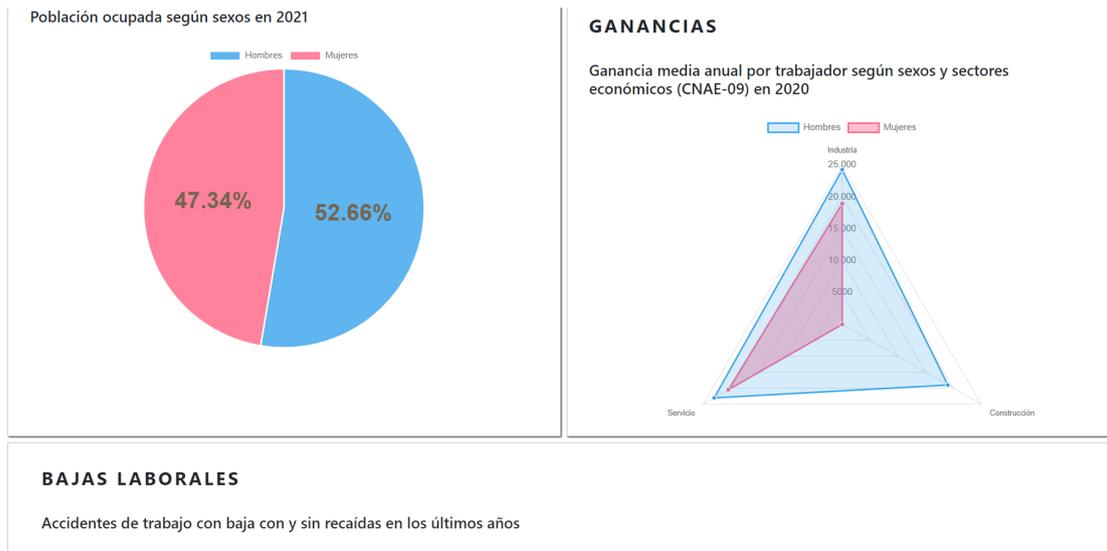


Figura 3.13: Página principal del dashboard de la aplicación web (1)

⁴Repositorio Frontend Web en GitHub: <https://github.com/alu0101228020/visualizaWeb.git>



BAJAS LABORALES

Accidentes de trabajo con baja con y sin recaídas en los últimos años

Figura 3.14: Página principal del dashboard de la aplicación web (2)



VISUALIZA
Dayana Armas Alonso

Figura 3.15: Página principal Dashboard de la aplicación web (3)

Sección en específico

Al acceder a una sección en específico, se muestran todos los gráficos existentes en relación a ese tema. Además, en la parte superior se dispone de ciertos filtros para que el usuario pueda personalizar los gráficos. Cuando se cambian los filtros, aquellos gráficos en los que afecte dichos filtros, se renderizan y se cargan nuevamente.

[← Volver al Dashboard](#)

POBLACIONES

Escoja los filtros que desee:

Sexo: Gráfico: Fecha: Año:

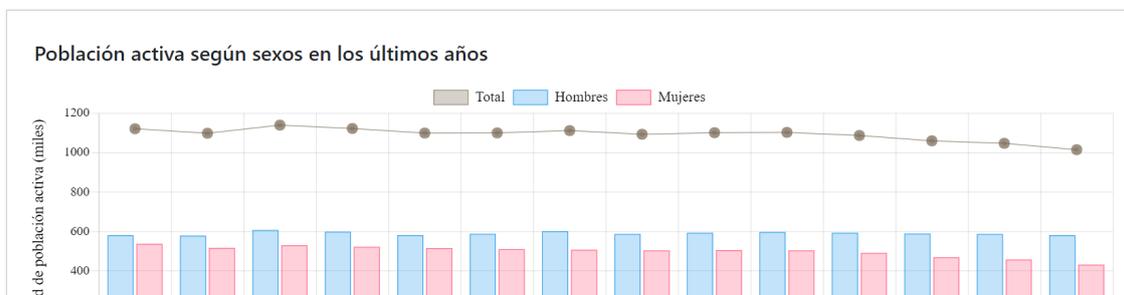
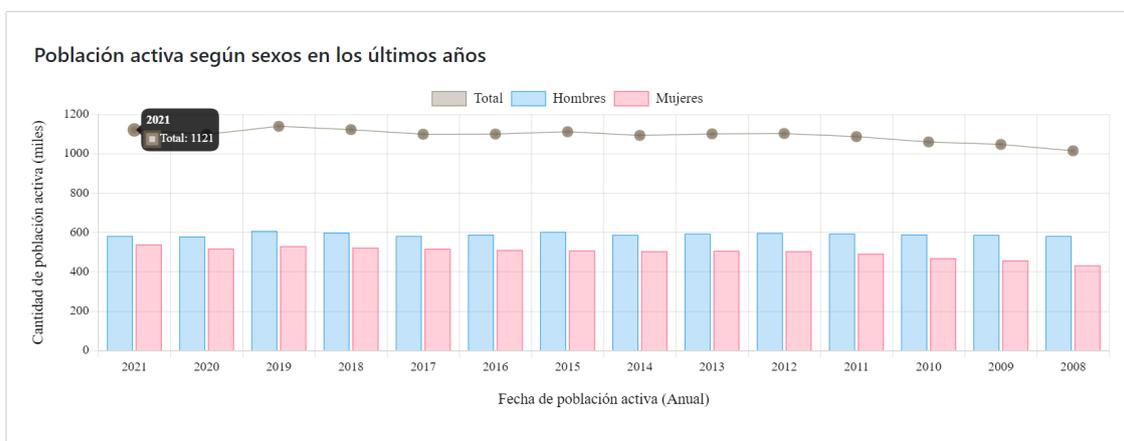


Figura 3.16: Sección de Poblaciones de la aplicación (1)

Todos los gráficos son interactivos y si el usuario desea mantener el foco en un punto en específico, aparece una leyenda con la información sobre ese punto.

Sexo: Gráfico: Fecha: Año:



Población activa, ocupada y parada en Canarias en los últimos años

Figura 3.17: Sección de Poblaciones de la aplicación (2)

3.5.2. Frontend móvil

El frontend de la aplicación móvil se ha realizado con el framework de Ionic y la integración de ReactJS. Además, el código fuente también está almacenado en un repositorio de GitHub⁵.

La estructura que sigue es bastante similar al frontend de la aplicación web a diferencia de las siguientes carpetas y ficheros:

⁵Repositorio Frontend Móvil en GitHub: <https://github.com/alu0101228020/visualizaMovil>

- **Android:** fichero de configuración que contiene los distintos archivos con la información de la aplicación en Android. Este se crea al configurar Android Studio en la aplicación.
- **ionic.config.json y capacitor.config.json:** se trata de archivos de configuración de la aplicación de Ionic.

Las distintas pantallas de la aplicación móvil se muestran de la siguiente manera:

Dashboard

Se puede observar que el dashboard es similar pero adaptado a versión móvil.



Figura 3.18: Página principal del dashboard de la aplicación móvil

Sección en específico

En la siguiente imagen se observa que existen elementos adaptados a la versión móvil como el desplegable de filtros que en este caso, al pulsar sobre él, se centra el foco y el filtro ocupa toda la pantalla.

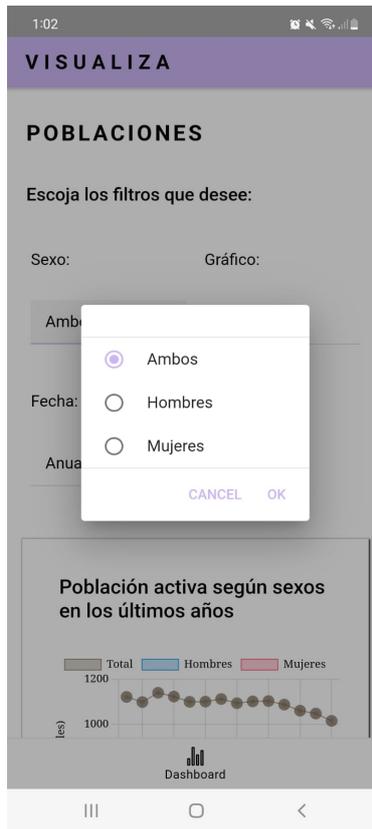


Figura 3.19: Filtro desplegado de la sección de Poblaciones de la aplicación móvil

También se observa que aparecen las leyendas al colocarse encima de algún punto de la gráfica.

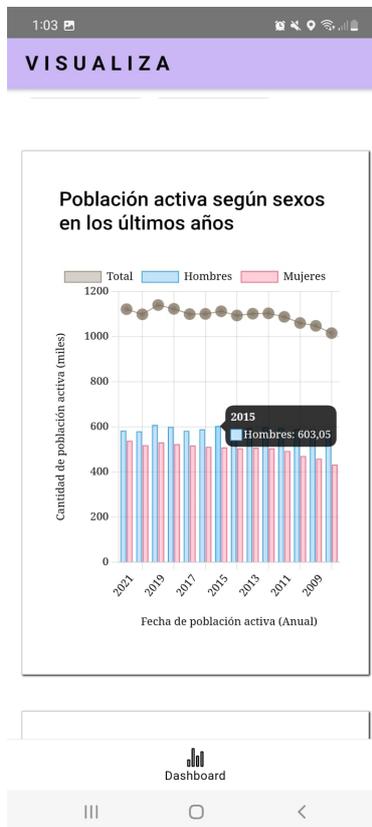


Figura 3.20: Sección de Poblaciones de la aplicación móvil

Cabe destacar que a pesar de la gran cantidad de datos, las gráficas se adaptan a la pantalla del dispositivo. Además, se puede colocar el móvil en posición horizontal permitiendo cambiar el modo de vista y de esta forma, se expande la gráfica para una mejor visualización.

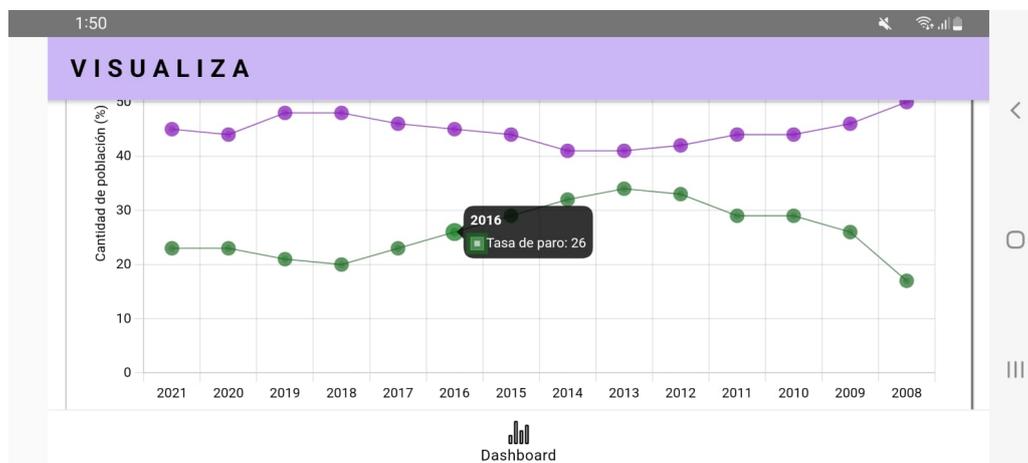


Figura 3.21: Vista horizontal de la aplicación móvil

3.6. Despliegue

En las siguientes secciones se tratará sobre los despliegues en ambas aplicaciones.

3.6.1. Entorno Heroku

Para realizar el despliegue del servidor backend y el frontend de la aplicación web, se ha utilizado Heroku. Esta plataforma tiene una versión gratis que permite desplegar las aplicaciones en un entorno pero si no se accede durante un tramo de tiempo a la página, los servidores se apagan para consumir menos recursos. Cabe destacar que si se accede a la página y los servidores están apagados, se debe esperar unos minutos para que el servidor backend y el servidor web se enciendan una vez que detecten actividad. En caso de que se obtuviese la versión de pago, los servidores no se apagarían. Si se desea visualizar la web, se dispone de la url de la misma⁶.

Ahora bien, para configurar Heroku se ha llevado a cabo la descarga del instalador para la plataforma de Windows [44]. Esta plataforma permite realizar el despliegue, una vez que se suba el proyecto de aplicación a un repositorio de GitHub de Heroku.

En primer lugar, se debe tener una cuenta registrada en Heroku. Una vez se accede a la página de Heroku, te permite crear el repositorio de GitHub de la aplicación donde se empujaron las carpetas y archivos de la aplicación local. Seguidamente, a través del comando de configuración `heroku create`, te permite configurar el entorno Heroku dentro de la aplicación local. Cuando se ejecuta dicho comando, también se crea un control remoto de git y se asocia el repositorio de GitHub anterior.

Una vez hecho esto, se finaliza la configuración de Heroku y por lo tanto, se empujan los cambios a la rama master de Heroku. Esto realiza la build que genera todo el código

⁶Url página web Visualiza: <https://visualizaappweb.herokuapp.com/>

objeto que utiliza Heroku para desplegar. Cada vez que sea necesario realizar un cambio, se debe empujar dichos cambios al repositorio de Heroku para actualizar la página.

Finalmente, cabe destacar que en la página de Heroku se puede administrar las aplicaciones desplegadas a través de una interfaz que permite ver la actividad sobre la aplicación y de esta manera, poder llevar un mejor control de la misma.

3.6.2. Entorno Android Studio

A continuación, para realizar el despliegue de la aplicación móvil, se debe obtener un APK generado por Android Studio que permite la instalación de la aplicación en los dispositivos móviles. Cuando se instala un proyecto en Ionic, también se configura Capacitor que es el tiempo de ejecución de la aplicación oficial de Ionic que facilita la implementación de aplicaciones web en plataformas nativas como iOS, Android y más. Debido a que se utiliza Capacitor, las aplicaciones Android se configuran y se administran a través de Android Studio.

Para comenzar a configurarlo, se realiza la build a través del comando *Ionic build*. A continuación, se crea el proyecto de Android con el comando *ionic cap add android*. Esto permite crear las carpetas android en la raíz del proyecto. Cada vez que se realice una compilación que actualice el directorio web a través del *ionic build*, se debe copiar esos cambios en el proyecto nativo con *ionic cap copy*.

Luego, se ejecuta *ionic cap open android*, que abre el proyecto nativo de Android en Android Studio. Finalmente, en este entorno se selecciona el dispositivo Android adjunto, luego se compila, se instala y finalmente, se inicia la aplicación en el dispositivo móvil. De esta manera, se simula el uso de la aplicación en este tipo de dispositivo aunque una vez que se genere el APK, se puede comprobar en un dispositivo real.

Ahora bien, para generar el APK, simplemente en el menú superior se selecciona la opción que permite generar dicho archivo. Esto descargará un fichero APK de la aplicación y por lo tanto, cuando se abre ese fichero en un dispositivo móvil, se instala la aplicación y a partir de ahí, se puede comenzar a utilizar en el dispositivo. Este fichero está disponible en la Nube⁷.

⁷Fichero APK de la aplicación: https://drive.google.com/file/d/1wdchwlHrHktnKD2VF6RHsWBB2RqvH6l_/view?usp=sharing

Capítulo 4

Estudio de viabilidad económica

Este apartado describe un análisis del estudio de la viabilidad económica sobre este proyecto. Se simulará el desarrollo de esta aplicación en versión web y móvil y además, se creará un plan de comercialización para el producto.

4.1. Desarrollo del proyecto

Para planificar el desarrollo del proyecto y conocer el presupuesto para llevarlo a cabo, se ha utilizado el programa ProjectLibre [45] que es un software de administración de proyectos de código abierto. El proyecto realizado está disponible en la nube para visualizarlo mejor¹.

En el desarrollo del proyecto se dará respuestas a las preguntas como: **¿Cuánto cuesta desarrollar este proyecto? ¿Cuánto tiempo tardaría en llevarse a cabo de forma profesional? ¿Cuándo se recuperaría la inversión inicial?** El proyecto se ha planteado en las siguientes cinco etapas:

- **Análisis:** en esta primera etapa se define el problema que se quiere resolver. Para ello, se establecen los requisitos, las funcionalidades y los costos que supondrá el proyecto. Además, se realizarán reuniones de coordinación para organizar a los distintos componentes del equipo y asegurar el buen desarrollo del proyecto.
- **Diseño:** como segunda etapa se estudiarán y se seleccionarán las tecnologías más adecuadas para el proyecto. Luego, se llevará a cabo el diseño de arquitectura para luego definir la estructura de la base de datos y del servidor backend. En relación al frontend, se realizará el diseño de las interfaces de la aplicación en ambas versiones.
- **Desarrollo e implementación:** una vez establecido los requerimientos así como las estructuras y diseños de la aplicación, se dispone a realizar la implementación de la base de datos, la implementación del backend para el procesamiento y tratamiento de los datos construyendo la API REST y finalmente, se lleva a cabo el desarrollo del frontend en ambas versiones.
- **Testeo:** en esta etapa se realizarán las pruebas necesarias para comprobar el correcto funcionamiento de los componentes.

¹**Proyecto de Project Libre:** https://drive.google.com/file/d/1Li1Qa5q-w5r14HpU4780q00Z_vt-tS67/view?usp=sharing

- **Producción:** en esta etapa se lleva a cabo el despliegue de la aplicación en los entornos considerados y se realizan pruebas sobre ello.

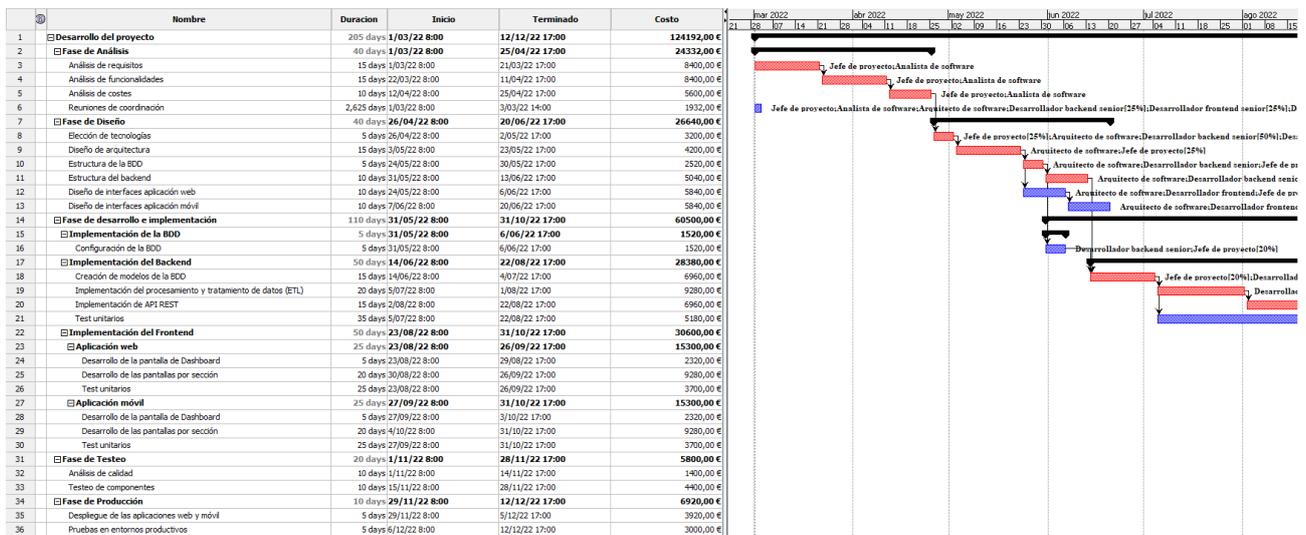


Figura 4.1: Diagrama de Gantt (1)

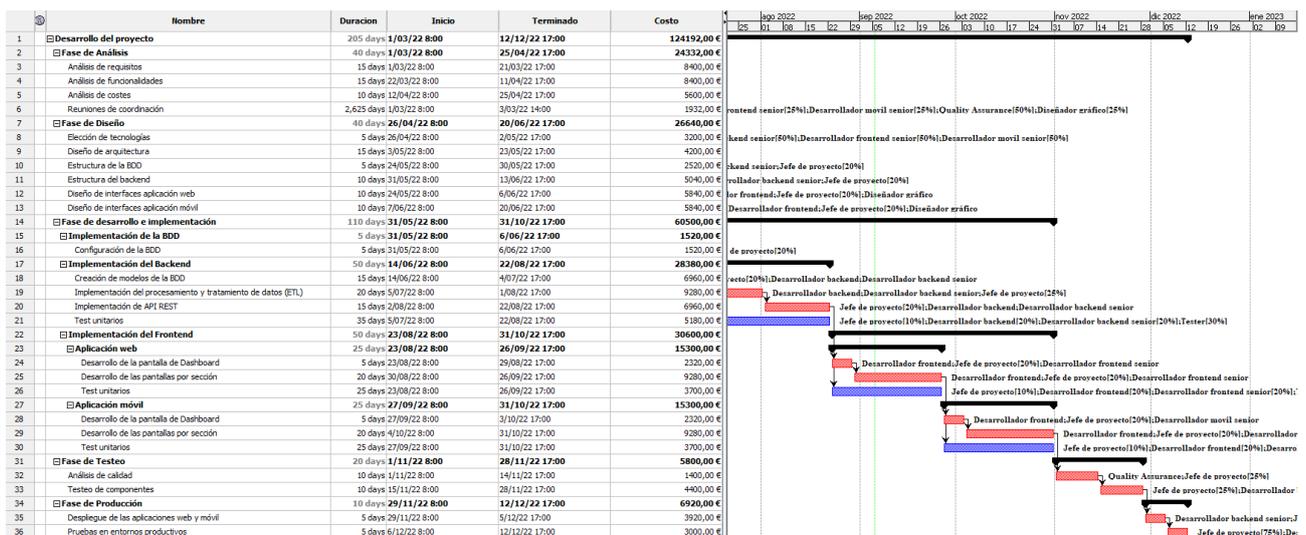


Figura 4.2: Diagrama de Gantt (2)

Una vez se termina esta última etapa, el producto puede empezar a ser comercializado. Para llevar a cabo las diferentes tareas se cuenta con el siguiente equipo:

- **Jefe de proyecto:** es la persona que gestiona el buen funcionamiento del proyecto con el fin de cumplir el objetivo definido. Sobre todo se encuentra activo en la fase de análisis.
- **Analista de software:** interviene en las primeras fases del proyecto donde se realizan las especificaciones de las necesidades que debe satisfacer la aplicación y define las especificaciones técnicas del producto correctamente.
- **Arquitecto de software:** es el que realiza el estudio de las posibles tecnologías, así como el software y el diseño de las arquitecturas que serán utilizadas.

- **Desarrollador backend:** es un programador especializado en la parte backend de la aplicación.
- **Desarrollador backend senior:** es un programador con bastante experiencia especializado en la parte backend de la aplicación.
- **Diseñador gráfico:** es el que diseña la interfaz de usuario en ambas versiones.
- **Desarrollador frontend:** es un programador especializado en la parte frontend de la aplicación.
- **Desarrollador frontend senior:** es un programador con bastante experiencia especializado en la parte frontend de la aplicación.
- **Desarrollador movil senior:** es un programador con bastante experiencia especializado en el desarrollo de aplicaciones móviles.
- **Tester:** se encarga de realizar el testeo de las partes de la arquitectura de la aplicación para comprobar su buen funcionamiento.
- **Quality Assurance:** se encarga de asegurar que la aplicación cumple los estándares de calidad establecidos.

Los salarios establecidos para cada uno de ellos son los siguientes:

- **Jefe de proyecto:** 40€/hora.
- **Analista de software:** 30€/hora.
- **Arquitecto de software:** 25€/hora.
- **Desarrollador backend:** 20€/hora.
- **Desarrollador backend senior:** 30€/hora.
- **Diseñador gráfico:** 20€/hora.
- **Desarrollador frontend:** 20€/hora.
- **Desarrollador frontend senior:** 30€/hora.
- **Desarrollador movil senior:** 30€/hora.
- **Tester:** 15€/hora.
- **Quality Assurance:** 15€/hora.

Tras los cálculos realizados de los costos de los recursos por tarea en el ProjectLibre, la duración del proyecto es de 205 días laborales, comenzando desde el 1 de marzo de 2022 y finalizando el 12 de diciembre de 2022, siendo una jornada laboral de 8 horas diarias de lunes a viernes. Finalmente, el costo total de este proyecto es de 124.192€.

4.2. Funcionalidades adicionales

La aplicación propuesta en este proyecto satisface totalmente las necesidades principales de un usuario aunque pueden complementarse con otras funcionalidades que son de interés para el usuario, como son:

- **Login:** esto permitiría al usuario logearse en la aplicación, eligiendo un plan de pago para poder hacer uso de la misma.
- **Distintas temáticas de interés:** la aplicación está enfocada al ámbito del empleo aunque se podría ampliar para abarcar otros temas de interés que se contemplan en la API de Canarias Datos Abiertos como son: ciencia y tecnología, comercio, educación, salud, demografía, etc.
- **Comparativa de gráficas entre provincias:** actualmente la aplicación esta centraliza para los conjuntos de datos de Canarias pero se podría ampliar añadiendo una sección de España donde se comparen ciertos datos entre distintas provincias. De esta manera, se podría comparar el estado y la evolución de Canarias respecto a las demás partes de España.
- **Comparativa de gráficas entre países:** también se podría añadir como funcionalidad ciertas gráficas comparativas entre países y así, no solo se tendría información en referencia al ámbito nacional sino también a nivel internacional.
- **Añadir informes de estudio:** podría realizarse informes de estudio que hubiesen hecho ciertos analistas sobre las gráficas y de esta manera, poder ofrecer mayor información al usuario.
- **Variación de idiomas:** el idioma que se está utilizando en la aplicación es el español, aunque se podría añadir una opción de cambio de idioma que permita traducir la página en el idioma en el que el usuario se sienta más cómodo.

Estas funcionalidades se caracterizan por ser de mayor interés para el usuario permitiendo obtener un mayor conocimiento y mejorando la comodidad del mismo. Esta aplicación inteligente con estas características potentes se comercializaría a gran escala y obtendría grandes beneficios.

4.3. Modelo de comercialización

Para conocer el modelo de comercialización que podría considerarse para esta aplicación, es necesario conocer el coste de mantenimiento que podría generar.

Los gastos de mantenimiento son orientativos y varían teniendo en cuenta lo siguiente:

- **Almacenamiento base de datos:** debido a que esto varía en función de la masa de usuarios y del contenido que se va añadiendo, se comienza por 55€/mes, luego 120€/mes y por último, 210€/mes.
- **Hosting:** se trata del dominio en el que se tiene la página web y normalmente se contrata por año, correspondiendo a 30€/mes.

- **Mantenimiento web y móvil:** permite arreglar bugs, actualizar contenido, mantener las licencias actualizadas y demás funcionalidades por lo que consta de 119€/mes.

Por lo tanto, inicialmente sería 204€/mes pero conforme se va necesitando más almacenamiento se llegaría a 269€/mes y finalmente, a 359€/mes.

Una vez conocemos los gastos, se puede establecer el modelo de comercialización. Esta aplicación se puede comercializar en base a dos planes de usuario.

Por un lado, se tiene el plan estándar que sería de 4,99€/mes donde se permitiría al usuario logearse y navegar por la aplicación para ver todo tipo de gráficas, pero no podría personalizar las mismas mediante filtros.

Por otro lado, se tiene el plan avanzado que sería de 8,99€/mes donde se permitiría al usuario realizar las mismas funcionalidades que en el plan estándar incluida la funcionalidad para personalizar las gráficas.

4.4. Punto de Retorno de Inversión (ROI)

Una vez que el proyecto se encuentra en el mercado y teniendo en cuenta la inversión inicial, se debe analizar los ingresos y los gastos de mantenimiento que pueden ir generándose.

Para ello, se ha llevado a cabo una aproximación de la cantidad de usuarios que se suscriben semanalmente dependiendo de los planes definidos anteriormente. Para ello, se ha realizado de manera exponencial la masa de usuarios, sumando el plan estándar 2 usuarios cada semana hasta la semana 24, es decir, pasados 6 meses, en el cual se duplica a 4 nuevas suscripciones semanales y luego pasadas las siguientes 24 semanas se vuelve a duplicar los usuarios hasta 6 suscripciones semanales continuando de esta manera hasta el final.

Por otro lado, en el plan avanzado se comienza con 1 usuario de más por cada semana hasta la semana 24 que se duplica y se añaden 2 usuarios semanalmente y luego, se va incrementando el número de usuarios de uno en uno cada 24 semanas.

A continuación, se muestra la siguiente tabla que se ha realizado en Microsoft Excel [46] con los cálculos²:

²Plantilla Microsoft Excel con los cálculos del ROI: https://docs.google.com/spreadsheets/d/1U70303FSDgc8DevdmK6FBoZwNXVZ_i14/edit?usp=sharing&ouid=102815563576964888959&rtpof=true&sd=true

Semanas	Plan estándar	Usuarios		Ingresos semanales	Ingresos vs Gastos		Ingresos - Gastos (inversión inicial + gastos semanales)	Gastos acumulados	Ingresos acumulados
		Total estándar	Plan avanzado		Total avanzado	Gastos semanales			
1	2	9,98	0	0				124396	9,98
2	4	19,96	1	8,99				124396	38,93
3	6	29,94	2	17,98				124396	86,85
4	8	39,92	3	26,97				124396	153,74
5	10	49,9	4	35,96		204	-124360,2	124600	239,6
6	12	59,88	5	44,95	104,83			124600	344,43
7	14	69,86	6	53,94	123,8			124600	468,23
8	16	79,84	7	62,93	142,77			124600	611
9	18	89,82	8	71,92	161,74	204	-124031,26	124804	772,74
10	20	99,8	9	80,91	180,71			124804	953,45
11	22	109,78	10	89,9	199,68			124804	1153,13
12	24	119,76	11	98,89	218,65			124804	1371,78
13	26	129,74	12	107,88	237,62	204	-123398,6	125008	1609,4
14	28	139,72	13	116,87	256,59			125008	1810,07
15	30	149,7	14	125,86	275,56			125008	2085,63
16	32	159,68	15	134,85	294,53			125008	2380,16
17	34	169,66	16	143,84	313,5	204	-122518,34	125212	2693,66
18	36	179,64	17	152,83	332,47			125212	3026,13
19	38	189,62	18	161,82	351,44			125212	3377,57
20	40	199,6	19	170,81	370,41			125212	3748,98
21	42	209,58	20	179,8	389,38	204	-121278,64	125416	4137,36
22	44	219,56	21	188,79	408,35			125416	4545,71
23	46	229,54	22	197,78	427,32			125416	4981,19
24	48	239,52	23	206,77	446,29			125416	5307,48
25	50	249,5	24	215,76	465,26	269	-119912,26	125685	5772,74
26	52	259,48	25	224,75	484,23			125685	6266,95
27	54	269,46	26	233,74	503,2			125685	6799,1
28	56	279,44	27	242,73	522,17			125685	7369,19
29	58	289,42	28	251,72	541,14	269	-117976,78	125954	7977,22
30	60	299,4	29	260,71	560,11			125954	8623,19
31	62	309,38	30	269,7	579,08			125954	9307,1
32	64	319,36	31	278,69	598,05			125954	10028,95
33	66	329,34	32	287,68	617,02	269	-115434,26	126223	10786,74
34	68	339,32	33	296,67	635,99			126223	11586,47
35	70	349,3	34	305,66	654,96			126223	12422,14
36	72	359,28	35	314,65	673,93			126223	13225,85
37	74	369,26	36	323,64	692,9				
38	76	379,24	37	332,63	711,87				
39	78	389,22	38	341,62	730,84	269	-114436,53	126492	14137,4
40	80	399,2	39	350,61	749,81			126492	15086,89
41	82	409,18	40	359,6	768,78			126492	16074,32
42	84	419,16	41	368,59	787,75			126492	17099,69
43	86	429,14	42	377,58	806,72	269	-108598	126761	18163
44	88	439,12	43	386,57	825,69			126761	19264,25
45	90	449,1	44	395,56	844,66			126761	20403,44
46	92	459,08	45	404,55	863,63			126761	21580,57
47	94	469,06	46	413,54	882,6	269	-104234,36	127030	22795,64
48	96	479,04	47	422,53	901,57			127030	24048,65
49	98	489,02	48	431,52	920,54			127030	25339,6
50	100	499,0	49	440,51	939,51	269	-101690,4	127300	26706,63
51	102	508,98	50	449,5	958,48			127300	27744,74
52	104	518,96	51	458,49	977,45			127300	28862,91
53	106	528,94	52	467,48	996,42			127300	30034,57
54	108	538,92	53	476,47	1015,39	269	-99551,48	127568	31363,14
55	110	548,9	54	485,46	1034,36			127568	32848,62
56	112	558,88	55	494,45	1053,33			127568	33509,101
57	114	568,86	56	503,44	1072,3			127568	34296,31
58	116	578,84	57	512,43	1091,27	269	-98436,09	127829	35199,09
59	118	588,82	58	521,42	1110,24			127829	36169,11
60	120	598,8	59	530,41	1129,21			127829	37204,07
61	122	608,78	60	539,4	1148,18			127829	38348,62
62	124	618,76	61	548,39	1167,15	269	-92476,99	127568	39501,01
63	126	628,74	62	557,38	1186,12			127568	40769,31
64	128	638,72	63	566,37	1205,09			127568	42156,61
65	130	648,7	64	575,36	1224,06			127568	43668,52
66	132	658,68	65	584,35	1243,03			127568	45310,57
67	134	668,66	66	593,34	1262,0	269	-89021,48	127300	46997,69
68	136	678,64	67	602,33	1280,97			127300	48734,66
69	138	688,62	68	611,32	1299,94			127300	50526,56
70	140	698,6	69	620,31	1318,91			127300	52378,27
71	142	708,58	70	629,3	1337,88	359	-78767,93	127927	54285,56
72	144	718,56	71	638,29	1356,85			127927	56244,61
73	146	728,54	72	647,28	1375,82			127927	58260,92
74	148	738,52	73	656,27	1394,79			127927	60339,09
75	150	748,5	74	665,26	1413,76	359	-74060,44	128286	62484,62
76	152	758,48	75	674,25	1432,73			128286	64693,66
77	154	768,46	76	683,24	1451,7			128286	66962,66
78	156	778,44	77	692,23	1470,67			128286	69297,66
79	158	788,42	78	701,22	1489,64	359	-62867,34	129004	71694,47
80	160	798,4	79	710,21	1508,61			129004	74149,82
81	162	808,38	80	719,2	1527,58			129004	76669,66
82	164	818,36	81	728,19	1546,55			129004	79249,66
83	166	828,34	82	737,18	1565,52	359	-55037,76	129004	81884,13
84	168	838,32	83	746,17	1584,49			129004	84578,66
85	170	848,3	84	755,16	1603,46			129004	87327,66
86	172	858,28	85	764,15	1622,43	359	-52729	129363	89134,66
87	174	868,26	86	773,14	1641,4			129363	91004,66
88	176	878,24	87	782,13	1660,37			129363	92942,66
89	178	888,22	88	791,12	1679,34			129363	94953,66
90	180	898,2	89	800,11	1698,31	359	-49938,43	129363	97034,66
91	182	908,18	90	809,1	1717,28			129363	99182,66
92	184	918,16	91	818,09	1736,25			129363	101392,66
93	186	928,14	92	827,08	1755,22			129363	103662,66
94	188	938,12	93	836,07	1774,19			129363	105998,66
95	190	948,1	94	845,06	1793,16	359	-44129,65	129363	108406,66
96	192	958,08	95	854,05	1812,13			129363	110884,66
97	194	968,06	96	863,04	1831,1			129363	113438,66
98	196	978,04	97	872,03	1850,07			129363	116064,66
99	198	988,02	98	881,02	1869,04			129363	118760,66
100	200	998,0	99	890,01	1888,01	359	-41470,44	129363	121534,66
101	202	1007,98	100	898,99	1906,98			129363	124383,66
102	204	1017,96	101	907,98	1925,95			129363	127294,66
103	206	1027,94	102	916,97	1944,92			129363	130264,66
104	208	1037,92	103	925,96	1963,89			129363	133291,66
105	210	1047,9	104	934,95	1982,86	359	-38376,35	129363	136381,66
106	212	1057,88	105	943,94	2001,83			129363	139531,66
107	214	1067,86	106	952,93	2020,8			129363	142748,66
108	216	1077,84	107	961,92	2039,77			129363	146029,66
109	218	1087,82	108	970,91	2058,74			129363	149381,66
110	220	1097,8	109	979,9	2077,71	359	-35206,38	129363	152802,66
111	222	1107,78	110	988,89	2096,68			129363	156291,66
112	224	1117,76	111	997,88	2115,65			129363	159845,66
113	226	1127,74	112	1006,87	2134,62			129363	163462,66
114	228	1137,72	113	1015,86	2153,59			129363	167140,66
115	230	1147,7	114	1024,85	2172,56	359	-32046,41	130440	170877,66
116	232	1157,68	115	1033,84	2191,53			130440	174681,66
117	234	1167,66	116	1042,83	2210,5			130440	178550,66
118	236	1177,64	117	1051,82	2229,47			130440	182482,66
119	238	1187,62	118	1060,81	2248,44	359	-28976,44	130440	186485,66
120	240	1197,6	119	1069,8	2267,41			130440	190557,66
121	242	1207,58	120	1078,79	2286,38			130440	194697,66
122	244	1217,56	121	1087,78	2305,35			130440	198904,66
123	246	1227,54	122	1096,77	2324,32			130440	203177,66
124	248	1237,52	123	1105,76	2343,29			130440	207516,66
125	250	1247,5	124	1114,75	2362,26	359	-25336,47	130440	211920,66
126	252	1257,48	125	1123,74	2381,23			130440	216390,66
127	254	1267,46	126	1132,73	2400,2			130440	220925,66
128	256	1277,44	127	1141,72	2419,17			130440	225525,66
129	258	1287,42	128	1150,71	2438,14	359	-22336,4	1	

Cabe destacar que se ha añadido ruido a los datos, es decir, que algunos datos no siguen la regla anterior ya que en la realidad, a la hora de lanzar la aplicación al mercado, no se suscriben los usuarios de manera exponencial cada semana. Existirán semanas donde se habrá incrementado mucho más la cantidad de usuarios mientras que en otras habrá un decrecimiento de los mismos.

Con el estudio realizado, se ha podido observar que el proyecto alcanzaría su punto ROI en la semana 90, es decir, que tardaría 22 meses y 2 semanas en recuperar la inversión inicial. A partir de ese punto, el proyecto generaría beneficios para sus inversores.

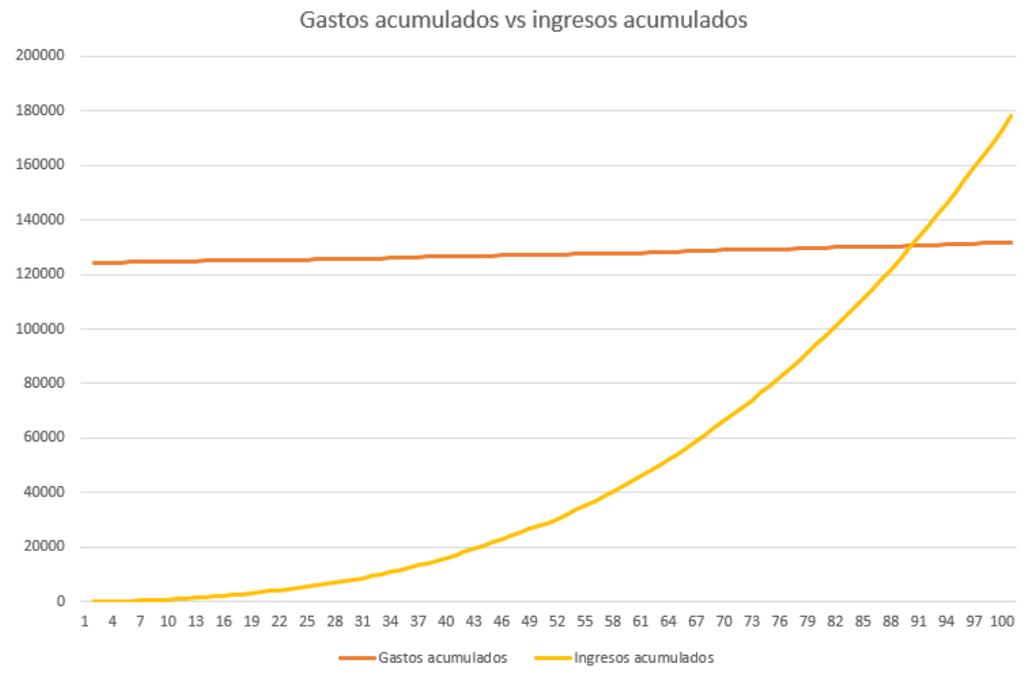


Figura 4.6: Punto de Retorno de Inversión

Capítulo 5

Conclusiones y líneas futuras

Una vez realizada con éxito dicha aplicación, se ha podido observar que se ha cumplido con todos los objetivos establecidos inicialmente. Además, he adquirido conocimientos técnicos en el desarrollo de aplicaciones web con React.js y en aplicaciones móviles con Ionic integrado también con React.js. Por otro lado, he mejorado ciertas habilidades en el uso de estructuras JSON a través del procesamiento de los datos para almacenarlos en la base de datos de MongoDB. Además, mediante la implementación de la API REST en el servidor se ha llevado a cabo una arquitectura simple, independiente y fácil de mantener. Finalmente, he podido realizar el despliegue del backend y de la aplicación web en entornos productivos como Heroku, además de poder obtener el APK para aplicaciones móviles a través de Android Studio.

También se ha podido llevar a cabo un estudio de la viabilidad económica del proyecto en el cual he aprendido las fases principales de las que está compuesta un proyecto, las cuales están desglosadas a su vez en tareas y recursos. Esto me ha permitido tener una visión clara sobre la realidad del proyecto, una vez que llegue a comercializarse. Además, he aprendido acerca del cálculo del ROI, lo que me ha permitido profundizar sobre los costos de mantenimiento de un proyecto y de la ganancia que puede llegar a producirse a través de los usuarios. Esto me ha hecho conocer el tiempo que se necesita para obtener los ingresos necesarios para alcanzar los gastos de la inversión inicial, incluyendo también los gastos de mantenimiento que se producen mensualmente.

Por otro lado, me gustaría destacar que en un futuro si se desea mejorar la aplicación se podrían realizar las funcionalidades adicionales propuestas en la sección 4.2. Esto permitiría mejorar la aplicación adaptándola a todo tipo de usuarios y además, aumentaría la satisfacción de los mismos.

Capítulo 6

Summary and Conclusions

Once this application has been successfully carried out, it has been possible to observe that all the initially established objectives have been met. In addition, I have acquired technical knowledge in the development of web applications with React.js and mobile applications with Ionic integrated also with React.js. On the other hand, I have improved certain skills in the use of JSON structures through the processing of data to store it in the MongoDB database. In addition, by implementing the REST API on the server, a simple, independent and easy to maintain architecture has been carried out. Finally, I have been able to deploy the backend and the web application in production environments such as Heroku, in addition to being able to obtain the APK for mobile applications through Android Studio.

It has also been possible to carry out a study of the economic viability of the project in which I have learned the main phases of which a project is made up, which are broken down into tasks and resources. This has allowed me to have a clear vision of the reality of the project, once it reaches the market. In addition, I have learned about the calculation of ROI, which has allowed me to delve into the maintenance costs of a project and the profit that can be produced through users. This has made me know the time it takes to obtain the necessary income to cover the initial investment costs, including also the maintenance costs that occur monthly.

On the other hand, I would like to point out that in the future, if you want to improve the application, you could carry out the additional functionalities proposed in the section 4.2. This would allow the application to be improved, adapting it to all types of users and, furthermore, it would increase their satisfaction.

Capítulo 7

Presupuesto

Además del estudio realizado sobre la viabilidad económica del proyecto, también se ha llevado a cabo un estudio del coste que ha supuesto realizar este proyecto de aplicación. El cálculo del tiempo invertido se ha realizado en base a la duración del proyecto que es de 300 horas, acorde con lo establecido en la Resolución del 21 de marzo de 2011, de la Universidad de La Laguna, por la que se publica el plan de estudios de Graduado en Ingeniería Informática. En este informe se establece que el Trabajo de Fin de Grado tiene una duración de 12 créditos ECTS y debido a que cada crédito representa un total de 25 horas, finalmente, la duración del proyecto abarca 300 horas.

Tareas	Duración	Coste
Análisis del problema y tendencia del mercado	20 horas	15 €/h
Estudio y selección de tecnologías	45 horas	15 €/h
Implementación del Backend	90 horas	20 €/h
Implementación del Frontend web y móvil	65 horas	20 €/h
Documentación del proyecto	80 horas	15 €/h
Total	300 horas	5.275 €

Tabla 7.1: Presupuesto del proyecto de Trabajo de Fin de Grado

Por lo tanto, para realizar este proyecto hicieron falta 300 horas con un coste de 5.275€.

Bibliografía

- [1] *A la espera de un Big Bang de datos*. Último acceso: 22/08/2022. Statista. [Online]. Available: <https://es.statista.com/grafico/17734/cantidad-real-y-prevista-de-datos-generados-en-todo-el-mundo/>
- [2] *Aplicaciones del big data en las empresas financieras*. Último acceso: 22/08/2022. Statista. [Online]. Available: <https://es.statista.com/grafico/8484/aplicaciones-del-big-data-en-las-empresas-financieras/>
- [3] *GRÁFICO | Paro en Canarias: así ha evolucionado desde el inicio de la pandemia*. Último acceso: 23/08/2022. Statista. [Online]. Available: https://www.eldiario.es/canariasahora/economia/grafico-paro-canarias-evolucionado-canarias-inicio-pandemia_1_8966247.html/
- [4] *La economía canarias en gráficos*. Último acceso: 23/08/2022. CEOE-Tenerife. [Online]. Available: <https://ceoe-tenerife.com/wp-content/uploads/2021/05/2020-La-Economia-Canaria-en-graficos.pdf>
- [5] *Big data market size revenue forecast worldwide from 2011 to 2027*. Último acceso: 24/08/2022. Statista. [Online]. Available: <https://www.statista.com/statistics/254266/global-big-data-market-forecast/>
- [6] *How choose a JS framework*. Último acceso: 27/08/2022. Medium. [Online]. Available: <https://medium.com/@akiva10b/how-to-decide-on-a-js-framework-53e9a6bb6f3a>
- [7] *node-cron*. Último acceso: 28/08/2022. npm. [Online]. Available: <https://www.npmjs.com/package/node-cron>
- [8] *Datos Abiertos de Canarias*. Último acceso: 23/08/2022. Canarias Datos Abiertos. [Online]. Available: <https://datos.canarias.es/portal/>
- [9] *Tres tendencias de datos y análisis que dominarán 2022*. Último acceso: 24/08/2022. CIO ESPAÑA. [Online]. Available: <https://www.ciospain.es/big-data/tres-tendencias-de-datos-y-analisis-que-dominaran-2022#:~:text=La%20firma%20de%20investigaci%C3%B3n%20Fortune,2%25%20entre%202021%20y%202028.>
- [10] *Tableau*. Último acceso: 24/08/2022. Tableau. [Online]. Available: <https://www.tableau.com/es-es>
- [11] *Microsoft Power BI*. Último acceso: 24/08/2022. Microsoft. [Online]. Available: <https://powerbi.microsoft.com/es-es/>

- [12] *Google Charts*. Último acceso: 24/08/2022. Google. [Online]. Available: <https://developers.google.com/chart>
- [13] *nodejs*. Último acceso: 25/08/2022. nodejs. [Online]. Available: <https://nodejs.org/es/>
- [14] *expressjs*. Último acceso: 25/08/2022. expressjs. [Online]. Available: <https://expressjs.com/es/>
- [15] *Spring Boot*. Último acceso: 25/08/2022. spring. [Online]. Available: <https://spring.io/projects/spring-boot/>
- [16] *PostgreSQL*. Último acceso: 25/08/2022. PostgreSQL. [Online]. Available: <https://www.postgresql.org/>
- [17] *MongoDB*. Último acceso: 25/08/2022. MongoDB. [Online]. Available: <https://www.mongodb.com/es>
- [18] *ReactJS*. Último acceso: 26/08/2022. ReactJS. [Online]. Available: <https://es.reactjs.org/>
- [19] *Angular*. Último acceso: 26/08/2022. Angular. [Online]. Available: <https://angular.io/>
- [20] *VueJS*. Último acceso: 26/08/2022. VueJS. [Online]. Available: <https://vuejs.org/>
- [21] *React Native*. Último acceso: 26/08/2022. React Native. [Online]. Available: <https://reactnative.dev/>
- [22] *Ionic*. Último acceso: 26/08/2022. Ionic. [Online]. Available: <https://ionicframework.com/>
- [23] *D3.js*. Último acceso: 26/08/2022. D3js. [Online]. Available: <https://d3js.org/>
- [24] *Chart.js*. Último acceso: 26/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/>
- [25] *react-chartjs-2*. Último acceso: 26/08/2022. react-chartjs-2. [Online]. Available: <https://react-chartjs-2.js.org/>
- [26] *Iniciativa de datos abiertos del gobierno de españa*. Último acceso: 26/08/2022. datos.gob.es. [Online]. Available: <https://datos.gob.es/es>
- [27] *GitHub*. Último acceso: 26/08/2022. GitHub. [Online]. Available: <https://github.com/>
- [28] *Visual Studio Code*. Último acceso: 27/08/2022. Visual Studio Code. [Online]. Available: <https://code.visualstudio.com/>
- [29] *Android Studio*. Último acceso: 27/08/2022. Android Studio. [Online]. Available: <https://developer.android.com/studio>
- [30] *Heroku*. Último acceso: 27/08/2022. Heroku. [Online]. Available: <https://dashboard.heroku.com/>
- [31] *Thunder Client*. Último acceso: 27/08/2022. Thunder Client. [Online]. Available: <https://www.thunderclient.com/>

- [32] *MongoDB Atlas*. Último acceso: 27/08/2022. MongoDB Atlas. [Online]. Available: <https://www.mongodb.com/es/atlas/database>
- [33] *pgAdmin*. Último acceso: 27/08/2022. pgAdmin. [Online]. Available: <https://www.pgadmin.org/>
- [34] *Stack Overflow*. Último acceso: 27/08/2022. Stack Overflow. [Online]. Available: <https://stackoverflow.com/>
- [35] *chartjs-plugin-datalabels*. Último acceso: 27/08/2022. ChartJS. [Online]. Available: <https://chartjs-plugin-datalabels.netlify.app/>
- [36] *Documentación de API Canarias Datos Abiertos*. Último acceso: 28/08/2022. CKAN. [Online]. Available: <https://docs.ckan.org/en/2.9/api/>
- [37] *Axios*. Último acceso: 28/08/2022. Axios. [Online]. Available: <https://axios-http.com/>
- [38] *Bar Chart de ChartJS*. Último acceso: 28/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/docs/latest/charts/bar.html>
- [39] *Doughnut and Pie Charts de ChartJS*. Último acceso: 28/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/docs/latest/charts/doughnut.html>
- [40] *Line Chart de ChartJS*. Último acceso: 28/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/docs/latest/charts/line.html>
- [41] *Polar Area Chart de ChartJS*. Último acceso: 28/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/docs/latest/charts/polar.html>
- [42] *Radar Chart de ChartJS*. Último acceso: 28/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/docs/latest/charts/radar.html>
- [43] *Mixed Chart Types de ChartJS*. Último acceso: 28/08/2022. ChartJS. [Online]. Available: <https://www.chartjs.org/docs/latest/charts/mixed.html>
- [44] *Instalador Heroku*. Último acceso: 29/08/2022. Heroku. [Online]. Available: <https://devcenter.heroku.com/articles/heroku-cli>
- [45] *ProjectLibre*. Último acceso: 29/08/2022. Project Libre. [Online]. Available: <https://www.projectlibre.com/>
- [46] *Microsoft Excel*. Último acceso: 29/08/2022. Microsoft. [Online]. Available: <https://www.microsoft.com/es-es/microsoft-365/excel>