



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

Trabajo de Fin de Grado

Tratamiento y visualización de datos
abiertos sobre la demografía en España

*Open data treatment and visualization on the Spain
demography*

Daniel Álvarez Medina

La Laguna, 12 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:

"Tratamiento y visualización de datos abiertos sobre la demografía de España"

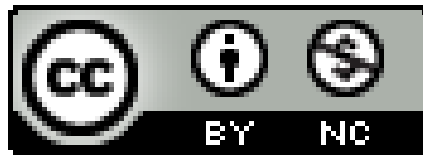
ha sido realizada bajo su dirección por D. **Daniel Álvarez Medina**, con N.I.F. 54.115.666-R.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 12 de septiembre de 2022

Agradecimientos

Quiero agradecer en primer lugar, a todos mis familiares y amigos que me han apoyado durante todos estos años, en los buenos y malos momentos. También a todas las personas que he conocido durante la carrera, que han hecho mi estancia en la Universidad más amena, y se han convertido rápidamente, en personas en las que puedo confiar. Por otro lado, agradezco a su vez a todos los compañeros de trabajo que me han ofrecido su apoyo y me han motivado en la recta final. Y finalmente a mi tutor Javier, por confiar en mí y guiarme pacientemente durante este proyecto final. Todos ellos han sido personas fundamentales durante esta etapa.

Licencia



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

Resumen

Durante los últimos años ha tenido lugar acontecimientos negativos que han afectado directamente a la demografía del país, como es el caso de la pandemia Covid-19. Sin embargo, existe un porcentaje de la población que no es plenamente consciente de las consecuencia de estos hechos, o simplemente se niegan a comprender la situación.

*Es por ello, que el presente proyecto tiene como objetivo realizar un prototipo **Full-Stack**, donde se traten, actualicen y visualicen automáticamente, datos referentes a la demografía del país, a fin de concienciar a la población mediante información verídica, y evitar se repitan acontecimientos como los sucedidos. Todo esto a su vez, se ha realizado siguiendo tendencias actuales como **Big Data** y **datos abiertos**.*

*La aplicación estará disponible tanto en la **web**, como en dispositivos **Android**. Y para obtener las distintas colecciones de datos acerca la demografía del País, se empleará la API pública de datos abiertos ofrecida por el Gobierno de España.*

*Finalmente se ha llevado una investigación de las tecnologías más adecuadas para el desarrollo del prototipo, así como un análisis de la competencia más relevante en el sector, y un estudio de la **viabilidad económica** de este proyecto en un ámbito profesional.*

Palabras clave: Demografía, Full-Stack, Big Data, Datos abiertos, Web, Android, API, España, Viabilidad económica

Abstract

In recent years, negative events have taken place that have directly affected the country's demographics, such as the Covid-19 pandemic. However, there is a percentage of the population that is not fully aware of the consequences of these events, or simply refuse to understand the situation.

*For this reason, this project aims to make a **Full-Stack** prototype, where data referring to the country's demography is automatically processed, updated and displayed, in order to raise awareness among the population through accurate information, and avoid repeating events such as those that occurred. All of this, in turn has been done following current trends such as **Big Data** and **open data**.*

*The app will be available on both the **web** and **Android** devices. And to obtain the different data collections about the demography of the Country, the public open data API offered by the Government of Spain, will be used.*

*Finally, a study of the most appropriate technologies for the development of the prototype has been carried out, as well as the most relevant competition in the sector, and a study of the **economic viability** of this project in a professional environment.*

Keywords: Demographics, Full-Stack, Big Data, Open Data, Web, Android, API, Spain, Economic viability

Índice general

1. Introducción	1
1.1. Definición del problema	1
1.2. Justificación	2
1.3. Tendencias del mercado	3
1.3.1. Big Data	3
1.3.2. Demografía	5
1.4. Objetivos	6
2. Estudio previo	8
2.1. Estudio del estado del arte	8
2.1.1. Power BI	8
2.1.2. Tableau	11
2.2. Arquitectura del prototipo	11
2.2.1. Backend	12
2.2.2. Frontend	13
2.2.3. Visualizador de datos	17
2.3. Base de datos	18
2.4. Entornos para despliegue	19
2.4.1. Dominios	20
3. Desarrollo de un prototipo	21
3.1. Selección de datos	21
3.2. Backend	22
3.2.1. Obtención de datos	22
3.2.2. Tratamiento de datos	22
3.2.3. Automatización de actualizaciones	23
3.2.4. Base de datos	25
3.2.5. API REST	26
3.3. Frontend	26
3.3.1. Llamadas a la API	27
3.3.2. Generación de gráficos	27
3.3.3. Conversión a Android	28
3.3.4. Pantallas	29
3.4. Despliegue	36
3.4.1. Creación y configuración de la máquina	36
3.4.2. Backend	36
3.4.3. Frontend	36

4. Estudio de viabilidad económica	37
4.1. Funcionalidades y herramientas	37
4.2. Desarrollo del proyecto	38
4.3. Comercialización	40
4.3.1. Fuentes de ingreso	40
4.3.2. Estudio del retorno de inversión (ROI)	41
5. Conclusiones y líneas futuras	43
6. Summary and Conclusions	44
7. Presupuesto	45

Índice de Figuras

1.1. Defunciones durante el Covid, comparada con el promedio de defunciones de otros años [1].	1
1.2. Pronóstico de ingresos del tamaño del mercado de big data en todo el mundo desde 2011 hasta 2027 (Expresado en miles de millones de dólares estadounidenses) [2].	4
1.3. Volumen de datos creados, capturados, copiados y consumidos en todo el mundo de 2010 a 2020, con pronósticos 2021-2025 (Expresado en Zettabytes) [3].	5
1.4. Los tres principales retos demográficos de MITECO [4].	6
2.1. Logo de Power Bi	8
2.2. Comparativa entre Power BI Pro y Power BI Premium (1)	9
2.3. Comparativa entre Power BI Pro y Power BI Premium (2)	10
2.4. Comparativa entre Power BI Pro y Power BI Premium (3)	10
2.5. Logo de Tableau	11
2.6. Logos de frameworks para el Backend	12
2.7. Logos de frameworks para el Frontend (1)	13
2.8. Logo de Flutter	14
2.9. Logos de frameworks para el Frontend (2)	14
2.10 Logo de React y React Native	15
2.11 Consumo de CPU de la aplicación realizada en Ionic	15
2.12 Consumo de CPU de la aplicación realizada React Native	16
2.13 Impacto energético de la aplicación realizada en Ionic	16
2.14 Impacto energético de la aplicación realizada en React Native	17
2.15 Logos de tecnologías que se van a usar junto con Ionic	17
2.16 Logos de herramientas para visualizar datos	18
2.17 Logos de bases de datos	19
2.18 Frameworks de despliegue	20
2.19 Páginas de dominios	20
3.1. Tiempos de referencia en node-cron	24
3.2. Nombre de las colecciones en la base de datos en MongoDB Atlas	25
3.3. Imágenes usadas para el logo de la aplicación	28
3.4. Logos de entornos de trabajo para el desarrollo móvil (2)	29
3.5. Página de inicio	30
3.6. Menú desplegable	30
3.7. Diagrama de barras sobre el total de la población española en 2021	31
3.8. Gráfico de líneas de la población en 2021 sobre el total, hombres y mujeres	31

3.9. Gráfico de radar de la población en Canarias sobre hombres y mujeres, durante distintos años	32
3.10 Gráfico circular de la población de hombres en 2003	32
3.11 Gráfico de área polar de la población de hombres y mujeres en Madrid, durante distintos años	33
3.12 Gráfico donut de la población total en 2021	33
3.13 Información del dataset	34
3.14 Menú desplegable oculto y pantalla de inicio	34
3.15 Gráfico de líneas en versión móvil vertical	35
3.16 Gráfico de líneas en versión móvil horizontal	35
4.1. Diagrama de Gantt (1)	40
4.2. Diagrama de Gantt (2)	40
4.3. Gráfica que muestra el punto de retorno de inversión	42

Índice de Tablas

3.1. Datasets elegidos del repositorio de datos del Gobierno de España	21
3.2. Especificación del número de entradas por cada dataset	25
3.3. Endpoints elegidos para cada dataset	26
7.1. Presupuesto del Trabajo de Fin de Grado	45

Capítulo 1

Introducción

1.1. Definición del problema

En los últimos años, la **demografía** del país se ha visto afectada negativamente por acontecimientos inesperados que han tenido un gran impacto, tanto en nuestras vidas como en nuestro planeta.

El suceso más notorio en este ámbito, es el caso de la pandemia Covid-19. Las consecuencias que trajo consigo fueron totalmente inesperadas, y no se tuvieron en cuenta hasta que los datos empezaron a hablar por sí solos, tal y como se aprecia en el siguiente gráfico.

DEFUNCIONES SEMANALES EN ESPAÑA (ENERO DE 2020-JULIO DE 2021) Y PROMEDIO DE DEFUNCIONES SEMANALES (2017-2019)

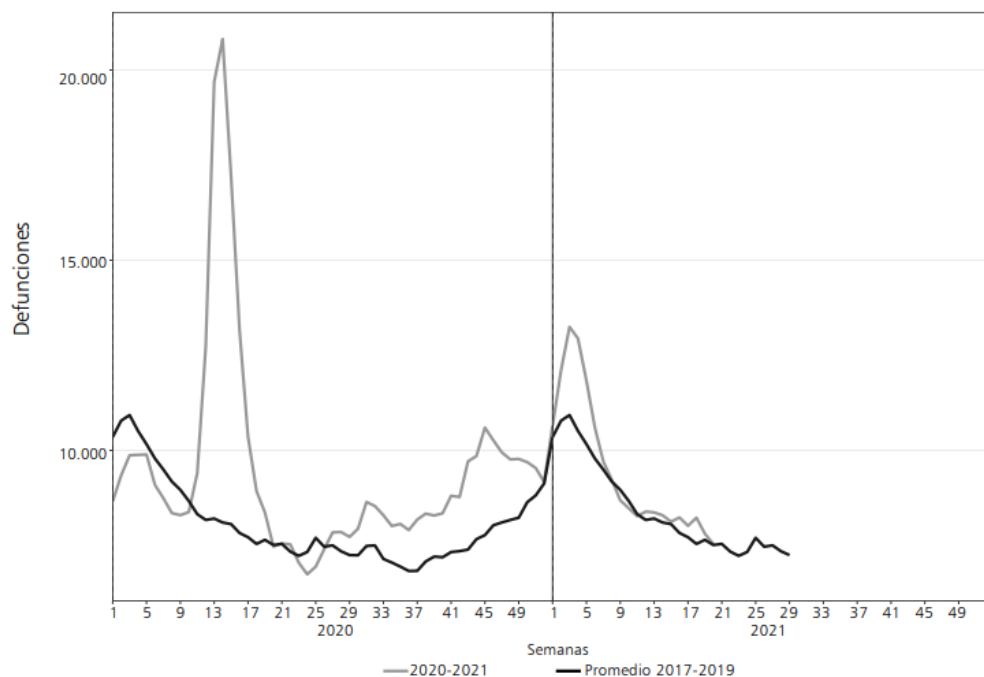


Figura 1.1: Defunciones durante el Covid, comparada con el promedio de defunciones de otros años [1].

Este desconocimiento de la gravedad del asunto en cierta medida surgió por la inexistencia de una infraestructura que pudiese manejar datos *fiabiles* y *accesibles*, con

el objetivo de compartirlos en *tiempo real*. Varios ejemplos de esta desinformación lo podemos ver reflejado en artículos del periódico **El País**, y tratan temas como; los negacionistas [5], las numerosas manifestaciones en contra de las medidas de seguridad o de la existencia del Covid [6], y los bulos o conspiraciones extendidos por anónimos y personajes públicos [7].

Sin embargo, esto no significa que en esta fase inicial no existiera información o datos al respecto, todo lo contrario. Nos encontramos en una era de avances tecnológicos, donde se ha generado una amplia variedad de datos en cantidades inmensas, desembocando en lo que hoy en día se conoce como **Big Data**¹. Este término ha adquirido una gran importancia, ya que gracias al *procesamiento de estos datos* se puede obtener información significativa para distintos ámbitos que ayuden a informar, además de contribuir a investigaciones o a la toma de decisiones.

Si durante los inicios de la pandemia, se hubiera empleado un mecanismo que tratase de una manera más adecuada el *Big Data* a la hora de analizar, representar y difundir la situación demográfica del país, la cantidad de desconocimiento y desconfianza hubiera sido menor.

Además, la demografía no es solamente útil para casos como estos, es una ciencia que se centra en el estudio de la población humana y su evolución, con el objetivo de anticipar posibles variaciones futuras.

1.2. Justificación

Para que acontecimientos similares no vuelvan a suceder, y para solventar este problema de comprensión de la importancia de la situación, surgen softwares de análisis de datos para recrear la **visualización** de dichos datos². Estas herramientas son importantes en la actualidad, debido a que aportan una solución a la cuestión planteada, a través del uso de elementos visuales como gráficas o tablas, para representar la información.

En relación a lo anterior, la visualización de datos se puede llevar a cabo en una amplia variedad de sectores, pero en la demografía adquiere una gran relevancia al tratarse de una ciencia de datos que, por medio de estudios estadísticos, obtiene información de la población humana en ámbitos cuantitativos y cualitativos.

A pesar de ello, la generación masiva de tantos datos referentes a la población, carecen de una representación visual y con actualizaciones frecuentes, que ayuden al entendimiento de los resultados. Por lo tanto, si se aplicase dicha representación a la demografía en España, se podrían emplear los resultados obtenidos para facilitar diversas labores como: orientar servicios u ofertas a un sector de la población, planificar políticas enfocadas al desarrollo social y económico de los pueblos, estrategias de empresas, analizar el envejecimiento de la sociedad, investigar los motivos del éxodo rural... Por consiguiente, la visualización de datos demográficos es un objetivo presente en muchos trabajos, con el fin de obtener beneficios ya sea a nivel de España o de empresa.

¹**Big data:** También conocido como macrodatos

²**Visualización de datos:** representación de datos en formato gráfico.

Es por todo lo previamente mencionado, que este presente proyecto, consiste en realizar una aplicación **Full-Stack**³ donde a través del tratamiento de los datos abiertos⁴ de la API⁵ ofrecida por el Gobierno de España[8], se ofrezca una visualización en tiempo real de los datos demográficos del país. Esto con la finalidad de proporcionar una manera intuitiva de analizar y comprender valores como tendencias, patrones y predicciones. Dicha aplicación contará con una interfaz web, así como un despliegue para dispositivos Android.

1.3. Tendencias del mercado

En la sección anterior se ha podido apreciar, que existen dos temas que cobran bastante relevancia en nuestro análisis de la situación del país: el Big Data y la demografía.

1.3.1. Big Data

Como se ha mencionado con anterioridad, el Big Data comprende a un volumen de datos de gran magnitud, variado, y que crece rápidamente. Para entender mejor las características y el comportamiento de este término, el analista **Doug Laney**, publicó en 2001 un artículo [9] donde define el Big Data en varios ámbitos denominados "**Las tres V**":

- **Volumen:** hace referencia a la cantidad de datos existentes a procesar. Cabe destacar que no se refiere solamente a *datos estructurados*⁶ como: información de clientes, resultados de encuestas, ventas realizadas, etc., sino también a datos no estructurados⁷ procedentes de fuentes como: redes sociales, sensores, aplicaciones, medios enriquecidos⁸, etc.
- **Velocidad:** continuamente se están generando datos, y cada vez más desde distintas fuentes de origen. Esto aumenta a la velocidad que se generan, y es por ello que las plataformas de Big Data deben de tener la capacidad de acceder a ellos, almacenarlos y tratarlos de una manera fluida, para mejorar la toma de decisiones.
- **Variedad:** está relacionado con parte de lo que hablamos en la sección de *Volumen*. Los datos están presentes en todo tipo de formatos, incorporando los datos no estructurados y semi-estructurados, como por ejemplo audios o vídeos. Estos datos requieren de un preprocesado adicional para poder extraer información de ellos.

Como se puede apreciar, las características del Big Data, hace que siempre esté en constante evolución. Tanto es así, que si apreciamos el siguiente gráfico, vemos que se

³**Full-Stack:** Desarrollo tanto en la parte del cliente y usuarios como del servidor (Backend y Frontend).

⁴**Datos abiertos:** También conocido como Open Data. Representan cualquier información digital de carácter accesible, reutilizable y con posibilidad de redistribución

⁵**API:** *Application Programming Interfaces*. Herramienta que permite que dos componentes de software puedan comunicarse entre sí mediante un conjunto de definiciones y protocolos.

⁶**Datos estructurados:** Datos almacenados de tal forma que la extracción de información de ellos es sencilla. Suelen integrarse en bases de datos

⁷**Datos no estructurados:** Son datos que carecen de una estructura identificable, es decir, que no se ajusta a ningún modelo de dato.

⁸**Medios enriquecidos:** hace referencia a medios audiovisuales como: textos decorados, audios o vídeos.

estima que para el año 2027, la cantidad de ingresos generados por el Big Data supere los 100 mil millones de dólares estadounidenses.

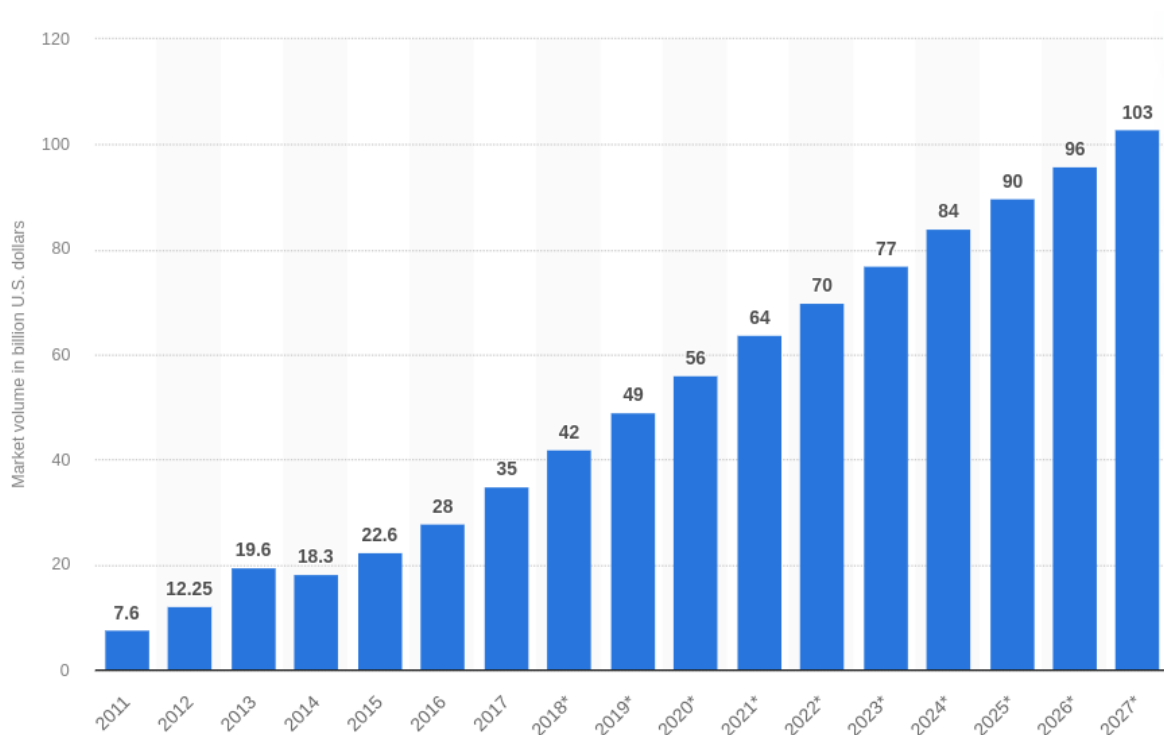


Figura 1.2: Pronóstico de ingresos del tamaño del mercado de big data en todo el mundo desde 2011 hasta 2027 (Expresado en miles de millones de dólares estadounidenses) [2].

Finalmente, no sólo se estima que la cantidad de ingresos vaya a aumentar notoriamente. También se han hechos pronósticos del volumen de información que será creado y consumido, alcanzando valores cercanos a los 200 *Zettabytes*⁹.

⁹**Zettabytes:** Equivale a equivale a 10^{21} bytes.

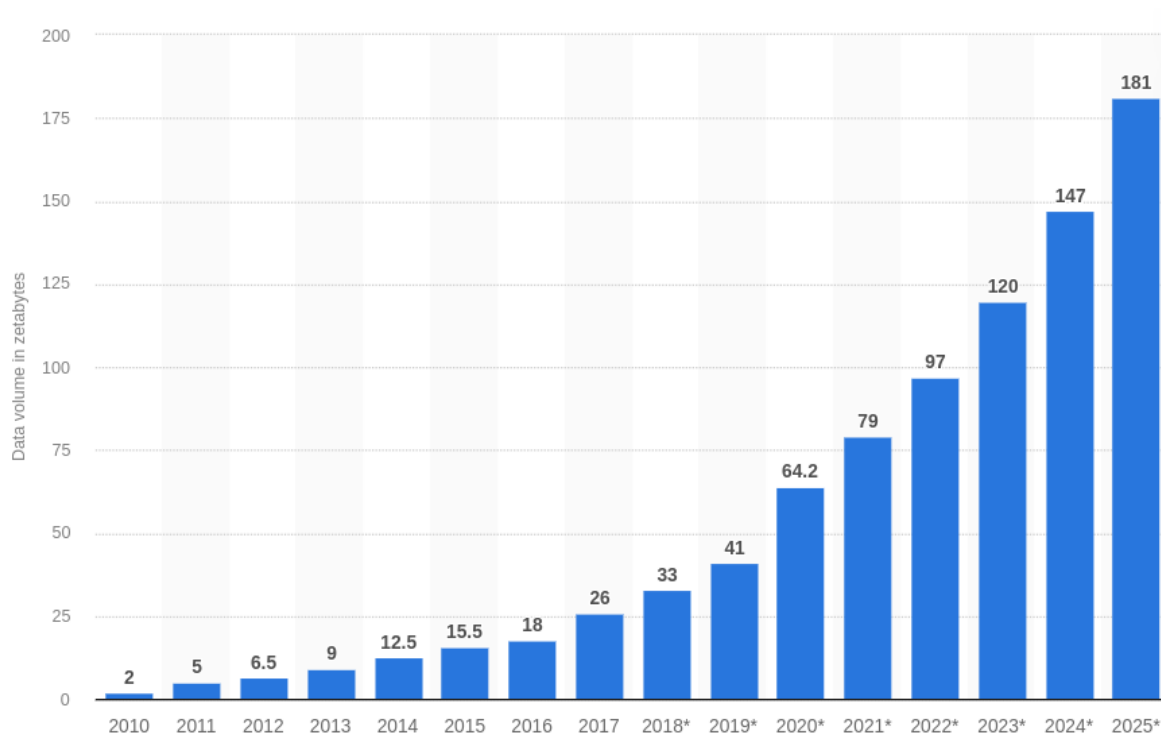


Figura 1.3: Volumen de datos creados, capturados, copiados y consumidos en todo el mundo de 2010 a 2020, con pronósticos 2021-2025 (Expresado en Zettabytes) [3].

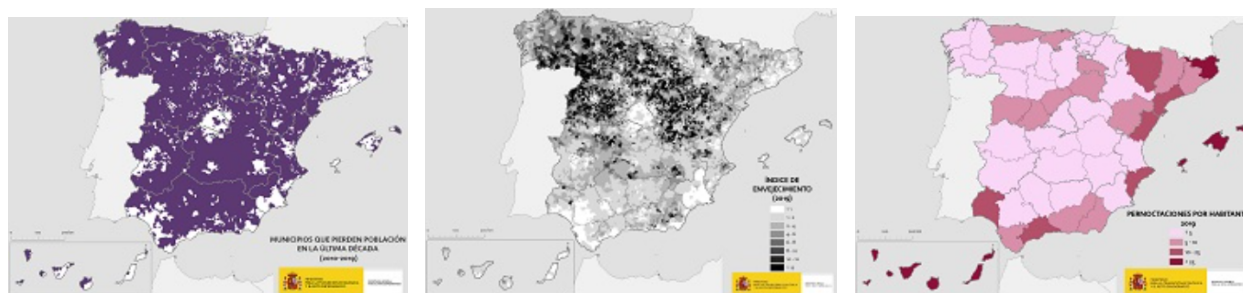
1.3.2. Demografía

A su vez, se ha mencionado previamente que la demografía es una ciencia que estudia al ser humano y su evolución, con el objetivo de analizar y predecir situaciones.

Analizando un artículo del *Ministerio para la Transición Ecológica y Reto Demográfico* (MITECO) [4], vemos que en España la demografía cuenta con aspectos positivos como; el aumento de la esperanza de vida y el crecimiento de la población en el siglo XXI. Sin embargo, también contamos con declives en algunos aspectos tales como: la baja natalidad, éxodo rural, etc.

Es por ello, que MITECO ha planteado en España un total de 130 medidas [10], orientadas a un conjunto de objetivos para conseguir una igualdad de oportunidades, reforzamiento de los vínculos rurales y urbanos, y vertebración territorial entre otras metas. En concreto, los principales desafíos de los retos demográficos son tres: **despoblación, progresivo envejecimiento y los efectos de la población flotante** ¹⁰.

¹⁰**Población flotante:** Población que reside en un territorio diferente a su lugar de residencia



(a) Despoblación

(b) El progresivo envejecimiento

(c) Los efectos de la población flotante

Figura 1.4: Los tres principales retos demográficos de MITECO [4].

Para apoyar estas medidas, en algunos casos se han recibido ayudas, como es el caso de 16,2 millones de euros para proyectos innovadores de transformación territorial y lucha contra la despoblación [11].

1.4. Objetivos

El objetivo de este estudio, por un lado es realizar una aplicación de visualización de datos demográficos para la web y dispositivos Android. Sin embargo, también se centra en solventar la duda de si es viable un proyecto de estas características.

Para lograr estos hitos, se han establecido una serie de objetivos:

- Análisis de hardware, herramientas de software y tecnologías disponibles para desarrollar el prototipo.
- Análisis del mercado para identificar la competencia y los productos de software más destacados referentes a la visualización de datos.
- Estudio de funcionalidades o características para que el prototipo destaque entre la competencia.
- Desarrollo de la arquitectura de la aplicación de visualización de datos:
 - Implementación del Backend¹¹:
 - Realizar llamadas a la API del Gobierno de España, para obtener los datasets con los que se van a trabajar.
 - Tratamiento de los datasets a fin de obtener los datos de la forma deseada.
 - Automatizar tareas periódicas para actualizar los datos de manera autónoma.
 - Almacenar en una base de datos la información tratada.
 - Crear API REST¹² para leer y consultar los datos almacenados en la base de datos.

¹¹**Backend:** Es la parte del servidor en un sitio web. Se encarga de que toda la lógica de una página web funcione.

¹²**REST:** Representational State Transfer, o Transferencia de Estado Representacional en español. Arquitectura con una serie de principios o restricciones, que se emplean para conectar varios sistemas basados en el protocolo HTTP.

- Implementación del Frontend¹³:
 - Realizar diseño de las distintas pantallas.
 - Realizar las distintas llamadas a la API REST alojada en el Backend, para obtener los datasets.
 - Implementar las diferentes gráficas para los datos.
 - Permitir al usuario poder escoger entre distintas características: gráficas, géneros, fechas o ubicaciones.
 - Despliegue de la aplicación web.
 - Conversión a Android mediante Capacitor.
 - Despliegue en dispositivos Android.
 - Comprobar el comportamiento de la aplicación.
- Desarrollo de un plan de negocio para comercializar el producto:
- Realizar un Diagrama de Gantt ¹⁴ , el cual incluya las tareas a realizar, sus duraciones, y los individuos y recursos asignados a cada una de estas tareas.
 - Estimar el costo del desarrollo del producto final.
 - Estudiar y diseñar un modelo de comercialización del producto.
 - Calcular el ROI¹⁵.

¹³**Frontend:** Es la parte cliente de un sitio web. Se centra en todo lo que ve el usuario y cómo interactúa

¹⁴**Diagrama de Gantt:** Herramienta gráfica de gestión del tiempo y recursos. Su objetivo es definir el tiempo y presupuesto previstos para realizar una serie de tareas y actividades.

¹⁵**ROI:** Siglas de *Retorno de Inversión* o *Return of Investment* en inglés. Calcula el punto donde las ganancias superan la inversión inicial realizada

Capítulo 2

Estudio previo

Para desarrollar un buen producto y ofrecer la mayor calidad posible, es de gran importancia llevar a cabo un estudio previo de las mejores tecnologías y herramientas que podamos usar en el mercado actual. Es por ello que esta sección recopila los resultados obtenidos en la investigación de alternativas y tecnologías disponibles.

2.1. Estudio del estado del arte

Actualmente existen opciones muy potentes a la hora de visualizar datos, en este apartado hablaremos de algunas de las más extendidas y lo que ofrecen.

2.1.1. Power BI

Se basa en una plataforma de análisis de datos enfocada al *business intelligence*¹, lanzada en 2011 y desarrollada por Microsoft [12]. Los usuarios podrán analizar, transformar y comprender los datos de una forma eficiente, crear informes y *dashboards*² interactivos, y visualizar datos.



Figura 2.1: Logo de Power Bi

Está disponible de manera gratuita en Windows, Mac, Android, IOS y en la web, siempre que se disponga de un correo profesional o educacional. A su vez, Power BI cuenta con varios planes y productos disponibles:

- **Power BI Desktop:** Es el plan básico y gratuito de Power BI. Con esta versión podrás acceder a Power BI desde todas las plataformas de manera limitada, pero no podrás trabajar en entornos compartidos entre otras desventajas. A pesar de esto, sigue siendo una herramienta bastante potente para la visualización de gráficos.

¹**Business intelligence:** Inteligencia empresarial. Conjunto de estrategias y tecnologías que gestionan la información de una empresa con el objetivo de mejorar la toma de decisiones estratégicas

²**Dashboard:** Panel de control que gestiona, monitoriza y controla información relevante para evaluar una empresa, negocio, campaña, producto...

- **Power BI Pro** [13] : Versión de pago con un precio de 8.40€ al mes por usuario, la cual incorpora todas las características de la versión gratuita. Con Power BI Pro desbloqueas: características en entornos móviles, más herramientas de análisis inteligentes y reforzadas, poder compartir informes y proyectos, y poder trabajar de manera colaborativa en la web.
- **Power BI Premium**: Este plan incluye todo lo disponible en la versión Pro, y tiene un coste de 16,90€ al mes por usuario. También se puede comprar por capacidad al mes, lo cual permite obtener el producto sin tener que generar licencias de usuario para los consumidores, los precios se encuentran a partir de 4.212,30€ al mes.

A continuación se muestra unas tablas procedentes del sitio web oficial [14], donde se presentan las diferencias entre Power BI Pro y Power BI Premium

Característica	Power BI Pro	Power BI Premium Por usuario	Power BI Premium Por capacidad
Colaboración y analítica			
Acceso de aplicaciones móviles	●	●	●
Publicación de informes para compartir y colaborar	●	●	
Informes paginados (RDL)		●	●
Consumo de contenido sin licencias por usuario			●
Informes locales con Power BI Report Server			●

Figura 2.2: Comparativa entre Power BI Pro y Power BI Premium (1)

Característica	Power BI Pro	Power BI Premium Por usuario	Power BI Premium Por capacidad
Preparación, modelado y visualización de datos			
Límite de tamaño del modelo	1 GB	100 GB	400 GB
Frecuencia de actualización	8/día	48/día	48/día
Conexión con más de 100 orígenes de datos	●	●	●
Cree informes y visualizaciones con Power BI Desktop ⁴	●	●	●
Integración de API y controles	●	●	●
Objetos visuales de IA	●	●	●
IA avanzada (análisis de texto, detección de imágenes, machine learning automatizado)		●	●
Conectividad de lectura /escritura con puntos de conexión XMLA		●	●
Flujos de datos (consulta directa, entidades vinculadas y calculadas, motor de cálculo mejorado)		●	●
Creación de datamarts		●	●

Figura 2.3: Comparativa entre Power BI Pro y Power BI Premium (2)

Característica	Power BI Pro	Power BI Premium Por usuario	Power BI Premium Por capacidad
Gobernanza y administración			
Seguridad y cifrado de datos	●	●	●
Métricas de creación, consumo y publicación de contenido	●	●	●
Gestión del ciclo de vida de las aplicaciones		●	●
Gestión de implementación en varias ubicaciones geográficas			●
Bring your own key (BYOK)			●
Disponibilidad de complementos con escalabilidad automática			●
Almacenamiento máximo	10 GB/usuario	100 TB	100 TB

Figura 2.4: Comparativa entre Power BI Pro y Power BI Premium (3)

Power BI también ofrece una amplia compatibilidad con otros productos y servicios de Microsoft, lo que permite a los usuarios integrar fácilmente sus datos con otros sistemas y aplicaciones.

A su vez, numerosas empresas emplean este producto, como podría ser: Nestlé, Nokia, Adobe, HP, etc. [15]

2.1.2. Tableau

Tableau [16] también se trata de una plataforma de análisis visual fundada en 2003, y posteriormente adquirida por la empresa *Salesforce* en 2019. El objetivo de esta tecnología es mejorar el análisis de datos, y ayudar a las personas a comprender mejor los datos mediante visualizaciones de estos.



Figura 2.5: Logo de Tableau

Cuenta con una versión de prueba gratuita, sin embargo, todas las opciones que ofrece Tableau son de pago [17]. Cuando compras una licencia, puedes elegir entre dos opciones de hospedaje:

- **Hospedado por Tableau con Tableau Cloud:** Es la manera más rápida y sencilla, y la más empleada por usuarios individuales.
- **Hospedado por la organización del comprador con Tableau Server:** Es la elección ideal si la organización cuenta con recursos de TI y desea hospedar Tableau en su propia infraestructura de TI.

Una vez seleccionado el hospedaje, existen varios planes disponibles, los cuales son tarifas anuales que se pagan mediante mensualidades:

- **Tableau Viewer:** Permite a los usuarios acceder a dashboards existentes. Está destinado a aquellas personas necesitan información de análisis para realizar sus tareas o labores, pero prefieren emplear contenido creado por otros usuarios. Cuenta con un precio de 15,00€ al mes, facturado anualmente en 180,00€.
- **Tableau Explorer:** Con esta opción, aparte de poder acceder a dashboards existentes, también permite la edición de estos. Su precio es de 42,00€ al mes, facturado anualmente en 504,00€.
- **Tableau Creator:** Es la versión más completa, con ella, además de lo disponible en la versión Explorer, los usuarios podrán conectar sus datos, crear visualizaciones y publicar sus dashboards en Tableau Desktop. Este plan es accesible por la cantidad de 70,00€ al mes, facturado anualmente en 840,00€.

Tableau también cuenta con otros productos e integraciones con otras herramientas, para facilitar el trabajo de manera segura. A su vez, dispone de una gran cantidad de clientes que emplean sus productos, algunos de ellos son: Nissan, Lenovo, Telefónica Tech, Red Hat, etc [18].

2.2. Arquitectura del prototipo

Las tecnologías o lenguajes a emplear varían dependiendo de la finalidad que se le vaya a dar. Por lo tanto, se han estudiado las alternativas en base a la utilidad que se esté buscando.

2.2.1. Backend

Existe una gran diversidad de opciones a la hora de realizar el Backend de una aplicación, algunas han perdurado desde sus inicios, y otras emergen con características novedosas. Sin embargo, la selección la he realizado en base a mis conocimientos y la popularidad de los frameworks³:

- **Spring Boot [19]:** Pertenece a **Spring**⁴, y el enfoque de este se basa en la velocidad, la simplicidad y la productividad. En concreto, Spring Boot facilita la creación de proyectos en Spring, y proporciona herramientas como: inyección de dependencias, microservicios, arquitectura asíncrona y sin bloqueos... Otro gran beneficio, es que funciona sobre una máquina virtual de Java (JVM), por lo que se puede emplear con los lenguajes de Java y Kotlin.
 - Ventajas: Multiplataforma, maneja bien la asincronía, comunidad de usuarios grande.
 - Desventajas: Potente pero complejo, poseo poca experiencia en Java y Kotlin, no poseo experiencia en Spring Boot.

- **Node.js [20]:** Es un entorno de ejecución de JavaScript orientado a eventos asíncronos. Es de código abierto, multiplataforma y se ejecuta en el lado del servidor. Emplea el motor de tiempo de ejecución JavaScript V8, el cual también es un motor open-source creado por Google, y cuya funcionalidad es convertir el proyecto en código de máquina. Esto permite que la computadora pueda ejecutar sin necesidad de interpretar el código primero, ignorando la compilación, y por lo tanto aumentando su velocidad. Para los proyectos en Node.js se puede usar el lenguaje de JavaScript o TypeScript.
 - Ventajas: Simple de entender, multiplataforma, gran cantidad de librerías, maneja bien la asincronía, comunidad de usuarios grande, poseo bastante experiencia en Node.js, JavaScript y TypeScript.
 - Desventajas: Inestabilidad entre versiones.

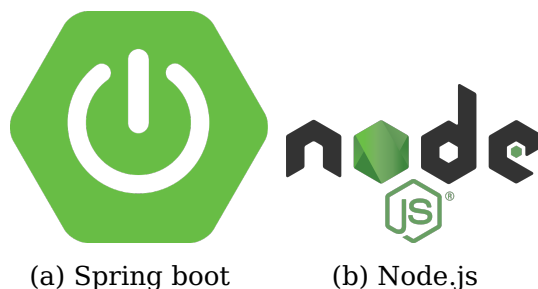


Figura 2.6: Logos de frameworks para el Backend

Lo que más buscaba entre las tecnologías, era poder manejar bien la asincronía y realizar el tratamiento de datos de una manera eficaz, ya que el proyecto cuenta con bastante peticiones a realizar. Ambas opciones cumplían con estos requisitos, así que me

³**Framework:** Entorno o marco de trabajo.

⁴**Spring:** Framework para el desarrollo de aplicaciones.

decanté por emplear **Node.js** junto a **JavaScript**, ya que es donde más experiencia tenía, además, JavaScript cuenta con una gran variedad de métodos que son de utilidad a la hora de trabajar con JSON⁵.

A su vez, conocía herramientas de JavaScript para fácilmente conectarme a la base de datos elegida, automatizar tareas y montar una API, por lo que estos conocimientos también me hicieron decantarme por la opción tomada.

2.2.2. Frontend

En cuanto al Frontend también existe una gran variedad de opciones, sin embargo, no todas son multiplataforma, lo cual es algo esencial ya que el proyecto tiene que funcionar tanto en web como en dispositivos móviles. Es por ello que las opciones que cumplían con estas características eran las siguientes:

- **Quasar [21]:** Es un framework basado en VueJS 3 [22] que se centra en la responsividad del proyecto, y con un solo código puedes generar aplicaciones web, móviles y de escritorio. En concreto, para generar aplicaciones móviles usa **Apache Cordova [23]**.
 - Ventajas: Con un solo código genero tanto la versión web como móvil, se emplea HTML, CSS y JavaScript o TypeScript
 - Desventajas: Aunque Apache Cordova sea potente y rinda bien, existen alternativas mejores, poseo poca experiencia con VueJS 3.0, tiene una comunidad pequeña en la que apoyarse.

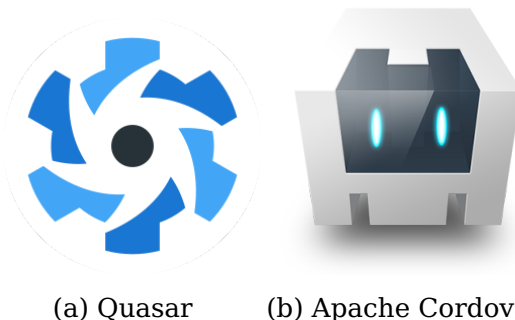


Figura 2.7: Logos de frameworks para el Frontend (1)

- **Flutter [24]:** Se trata de una tecnología de código abierto creada por Google, usada con el propósito de realizar aplicaciones multiplataforma compiladas de forma nativa a partir de una única base de código. El lenguaje que usa es Dart.
 - Ventajas: Con un solo código genero tanto la versión web como móvil, está respaldada por la gran empresa Google, la aplicación generada es nativa.
 - Desventajas: No poseo experiencia con el lenguaje de Dart, tiene pocas bibliotecas de terceros.

⁵**JSON:** JavaScript Object Notation. Formato de texto sencillo para el intercambio de datos, está basado en la notación de objetos de JavaScript



Figura 2.8: Logo de Flutter

- **Ionic [25]:** Este framework se basa en un conjunto de herramientas móviles de código abierto para crear experiencias de aplicaciones web y nativas multiplataforma de alta calidad, únicamente generando un único código para el proyecto. Ionic se puede emplear con Angular, VueJS o React. Para la conversión nativa a móvil usa **Capacitor [26]**, que a diferencia de Cordova con Quasar, Capacitor está realizado por los mismos creadores que Ionic, lo que favorece la compatibilidad y la optimización.
 - Ventajas: Con un solo código genero tanto la versión web como móvil, poseo experiencia con las distintas tecnologías que puede usar Ionic, la tecnología para convertir a móvil ha sido diseñada por los mismo creadores de Ionic, ha sido usada por grandes empresas como Huawei, Microsoft o Burger King.
 - Desventajas: En algunos casos concretos puede que no rinda tanto como una aplicación nativa.

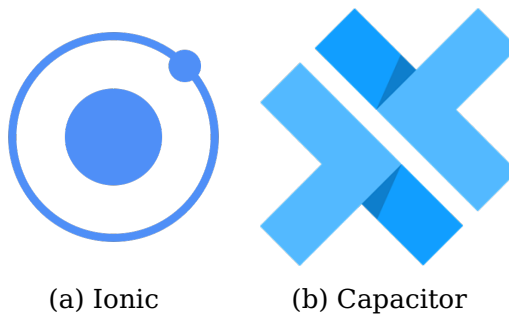


Figura 2.9: Logos de frameworks para el Frontend (2)

- **React y React Native [27, 28]:** Ambos fueron creados por Facebook. React es un framework diseñado para desarrollo web, mientras que React Native para desarrollo móvil. Los dos usan una sintaxis similar, por lo que a la hora de pasar de React a React Native, habría que solamente cambiar parcialmente el código. Emplea el lenguaje JSX, el cual es similar a JavaScript.
 - Ventajas: Son tecnologías apoyadas por Facebook, tienen una gran comunidad.
 - Desventajas: No poseo experiencia con React y React Native, tendría que hacer dos versiones del código, no todas las librerías de React están para React Native.

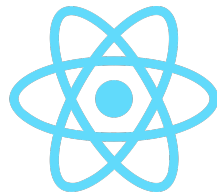


Figura 2.10: Logo de React y React Native

En un primer lugar, la tecnología que había decidido era **Quasar** ya que quería aprender Vue 3 y solo necesitaba una única base de código. Cabe destacar que cuando decidí usar esta tecnología, aún desconocía **Ionic**, sin embargo, cuando lo descubrí observé que tenía una mejor comunidad y usaba la herramienta Capacitor, la cual era más eficiente que Cordova [29].

A su vez, analizando Ionic frente a la alternativa popular, React Native, encontré un artículo [30] en el cual se hacía una comparativa entre ambos frameworks. En dicho escrito, en cuanto a experiencia de usuario, ambas alternativas ofrecían resultado similares. En cambio, en cuanto al consumo de la aplicación, en la siguiente figura 2.11, podemos observar que Ionic, tiene un consumo de CPU bastante bajo con un pico máximo del 10 %, ya que emplea *Just In Time (JIT) Compilation*⁶ .

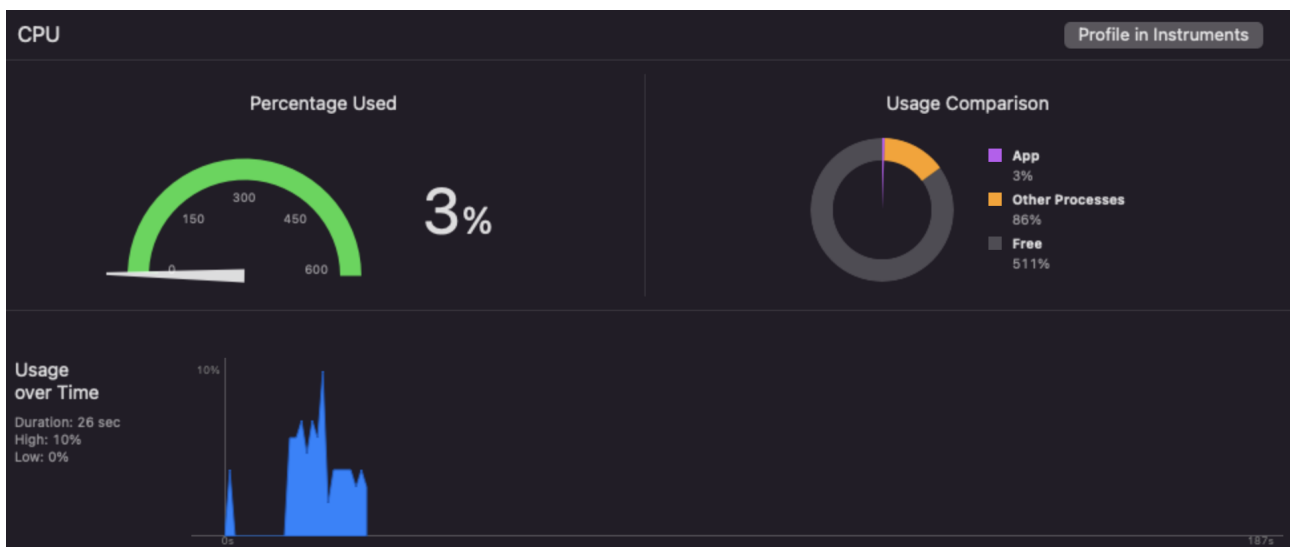


Figura 2.11: Consumo de CPU de la aplicación realizada en Ionic

Sin embargo, la figura 2.12 muestra el consumo de CPU en React Native, aquí podemos comprobar que tiene unos valores bastantes más elevados que en Ionic con un pico máximo del 199 %.

⁶**Just In Time Compilation:** Es una manera de ejecutar código que implica la compilación, pero durante la ejecución de un programa, es decir, en tiempo de ejecución en lugar de previamente.

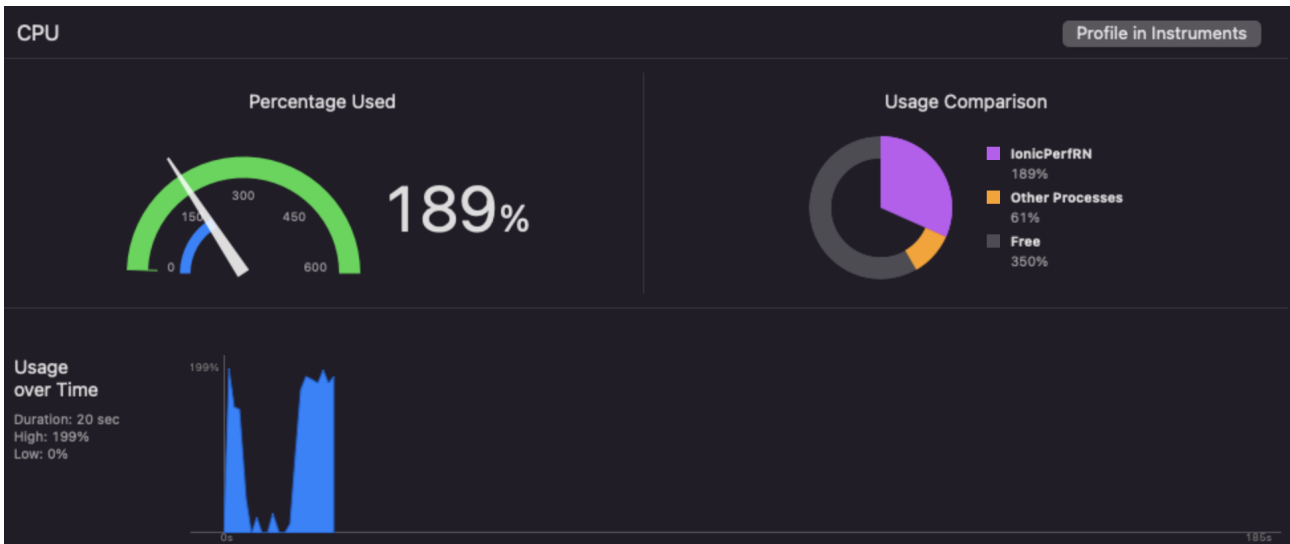


Figura 2.12: Consumo de CPU de la aplicación realizada React Native

Y esto no se limita solo a la CPU, en el artículo también se compara el impacto energético entre ambas alternativas, donde en la figura 2.13, se aprecia que Ionic cuenta con un alto consumo energético.



Figura 2.13: Impacto energético de la aplicación realizada en Ionic

Mientras que si nos fijamos en la figura 2.14, vemos que React Native posee un consumo muy alto. Esto implica que si se usa la aplicación durante un largo periodo de tiempo, la batería se consumiría antes si el prototipo es desarrollado en esta tecnología.



Figura 2.14: Impacto energético de la aplicación realizada en React Native

Por todos estos motivos, finalmente decidí emplear **Ionic** en conjunto a **Vue** y **Typescript** en el Frontend.

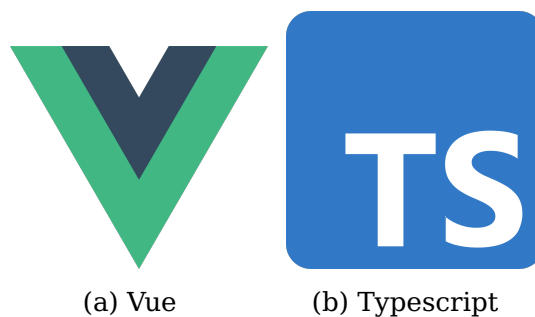


Figura 2.15: Logos de tecnologías que se van a usar junto con Ionic

2.2.3. Visualizador de datos

Una vez decidido el framework a usar en el Frontend, lo siguiente es estudiar la herramienta a emplear para visualizar los datos. Entre las alternativas a elegir, estaban las siguientes.

- **Chart.js [31]:** Es una librería open source⁷ de JavaScript, la cual cuenta con varios gráficos bastante personalizables y fáciles de usar.
 - Ventajas: Sencillo de usar, ofrece variedad de gráficos a emplear, tiene una extensión para Vue [32].

⁷Open source: Código abierto.

- Desventajas: Existen bastantes gráficas que no se encuentran entre las disponibles, habría que usar extensiones si se quieren aplicar dichos gráficos al proyecto.
- **D3.js [33]:** Es una biblioteca de JavaScript para manipular documentos basados en datos, empleando HTML, SVG y CSS.
 - Ventajas: No se limita sólo a gráficas, puedes crear otros elementos, puedes diseñar las gráficas a tu manera.
 - Desventajas: Hacer una gráfica simple lleva bastante tiempo y código, es complejo de usar.

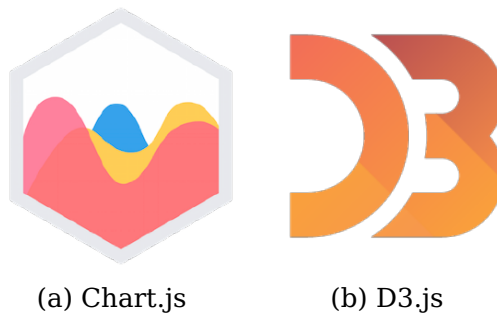


Figura 2.16: Logos de herramientas para visualizar datos

La opción elegida ha sido **Chart.js**, ya que es más simple de usar que D3.js, y se puede realizar muchos gráficos con esta herramienta. Inclusive, con las extensiones disponibles, se pueden llevar a cabo aún más variedad de gráficos.

2.3. Base de datos

Para almacenar toda la información generada, necesitamos una base de datos, entre las opciones se encuentran las siguientes:

- **MongoDB o MongoDB Atlas [34, 35]:** Se trata de una base de datos basada en documentos⁸ (No relacional), la cual es flexible y fácilmente escalable. MongoDB se trata de una base de datos local, mientras que MongoDB Atlas es una base de datos en la nube. Se basan en colecciones⁹, y los documentos¹⁰ están en formato BSON, una representación binaria de JSON.
 - Ventajas: Sencillo de usar, tengo experiencia con la tecnología, rápida, flexibilidad a la hora de crear las colecciones, se puede emplear la base de datos en local o en la nube gratuitamente.
 - Desventajas: Es más costoso realizar consultas¹¹ elaboradas ya que no es una base de datos relacional, tiene opciones de pago.

⁸Véase: https://es.wikipedia.org/wiki/Base_de_datos_documental

⁹**Colección:** Conjunto de documentos de una base de datos

¹⁰**Documento:** En el ámbito de base de datos, hace referencia a la información almacenada en una colección, pero que no tiene por qué tener una estructura definida (Algo que si sucede en las bases de datos relacionales).

¹¹**Consulta:** Sentencia que sirve para obtener información de la base de datos.

- **Firestore [36]:** Plataforma en la nube diseñada por Google, la cual se puede integrar fácilmente con muchos servicios como Google Ads, Google Play, Slack... Es una base de datos no relacional basada en colecciones y documentos. Además cuenta con una sección en la que puedes administrar los usuarios que se registren en la aplicación.
 - Ventajas: Flexibilidad a la hora de crear las colecciones, facilita el manejo de usuarios.
 - Desventajas: Es más costoso realizar consultas elaboradas ya que no es una base de datos relacional, las opciones gratuitas están bastante limitadas, no es de código abierto.

- **MySQL [37]:** Es un sistema de gestión de bases de datos relacional ¹², propiedad de Oracle y de código abierto.
 - Ventajas: Escalable, seguro, se pueden realizar búsquedas más complejas.
 - Desventajas: No es tan flexible como las bases de datos no relacionales, parte de las utilidades de MySQL no están documentadas.



Figura 2.17: Logos de bases de datos

Tras analizar las opciones, la base de datos elegida fue **MongoDB Atlas**. Era la tecnología que más había empleado, y aparte, tiene una extensión para Node.js bastante útil (mongoose [38]). A su vez, es bastante flexible a la hora de crear las colecciones, algo en lo que MySQL no es tan bueno. Y finalmente descarté Firebase, ya que aunque eran muy parecidas, no tenía tanto conocimiento de esta herramienta, y no necesitaba administrar usuarios.

2.4. Entornos para despliegue

Finalmente, todo el código desarrollado debe de alojarse en algún servidor, donde las alternativas barajadas han sido las siguientes:

- **Heroku [39]:** Se trata de una plataforma de servicios en la nube (PaaS), que permite a los usuarios construir y desplegar aplicaciones en la nube de manera gratuita. Heroku ejecuta sus aplicaciones en contenedores ligeros de Linux, llamados "dynos". La versión gratuita ofrece entre 550 y 1000 horas de uso al mes, 512MB de RAM, y el contenedor se apaga a los 30 minutos de inactividad.

¹²Véase: https://es.wikipedia.org/wiki/Base_de_datos_relacional

- Ventajas: Gratuito, diseño y configuración eficiente.
 - Desventajas: No tengo experiencia con Heroku, cantidad de horas limitadas al mes, limitación en las características de los contenedores, sólo soporta distribuciones de Linux.
- **DigitalOcean [40]:** Se trata de un proveedor de servicios en la nube, el cual ofrece una solución de alojamiento simple y segura. No es gratuito, pero DigitalOcean ofrece 100\$ durante 2 meses, además, al tener una cuenta de estudiante en Github, también te ofrece 100\$ gratis durante un año. Las máquinas de DigitalOcean se denominan droplets, y estas ofrecen distintas distribuciones y características a elegir para el servidor, en función del dinero disponible.
- Ventajas: Poseo experiencia con DigitalOcean, bastante personalización de los droplets, 100\$ gratuitos durante un año con el programa de Github Students.
 - Desventajas: Es de pago, sólo soporta distribuciones de Linux.

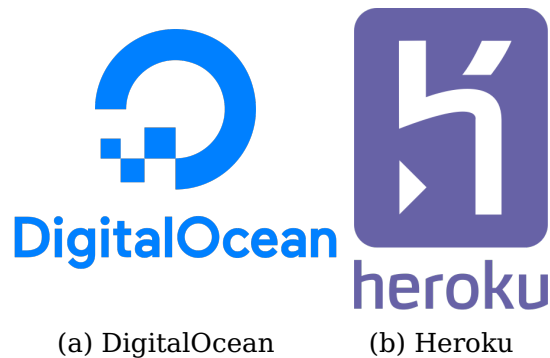


Figura 2.18: Frameworks de despliegue

Debido a que dispongo de una cuenta de estudiante de Github, elegí la opción de **DigitalOcean**, ya que me ofrecía una mayor personalización de los servidores, y lo había empleado previamente.

2.4.1. Dominios

Ya que poseo una cuenta de estudiante de Github, también tengo a mi disposición, varias páginas donde puedo seleccionar un dominio gratuito durante un año, dichas páginas eran: **Namecheap [41]** y **Name.com [42]**.



Figura 2.19: Páginas de dominios

Ambas ofrecían un dominio y certificado SSL [43] gratuito durante un año, y DigitalOcean cuenta con instrucciones de uso para cada una de estas páginas. Sin embargo elegí **Namecheap**, ya que la guía de pasos a seguir [44] en Name.com estaba desactualizada.

Capítulo 3

Desarrollo de un prototipo

En este capítulo veremos todas las tareas llevadas a cabo para realizar el prototipo. Cabe destacar que el nombre de la aplicación es **VDDE**, un acrónimo de **Visualización de Datos Demográficos en España**. Además, se puede apreciar el código en detalle, accediendo al repositorio de Github [45].

3.1. Selección de datos

Como ya se ha comentado previamente, todos los datos que se van a visualizar han sido extraídos del repositorio de datos abiertos procedente del Gobierno de España. Dicho sitio cuenta con una **API** a la cual se le puede realizar peticiones HTTP, y de esta manera, obtener los datasets necesarios.

Dentro de la sección de demografía, se han buscado datasets que cuenten con algún tipo de actualizaciones periódicas, ya que un objetivo del prototipo es actualizar los datos de manera automática. Teniendo esto en cuenta, finalmente se han elegido cuatro conjuntos, los cuales son los siguientes:

Dataset	Fuente	Frecuencia de actualización
Población española	INE	Semanal
Índice de natalidad	INE	Anual
Índice de fecundidad	INE	Anual
Tasa de mortalidad infantil	INE	Anual

Tabla 3.1: Datasets elegidos del repositorio de datos del Gobierno de España

Cabe destacar que los datasets escogidos engloban todas las comunidades autónomas, es decir, no se centra en una concreta.

El prototipo planteado no solo se limita a obtener datos, también genera otros nuevos. En concreto, generamos dos fuentes de datos nuevas, gracias a la información proporcionada por el Gobierno de España:

- **Cantidad de nacimientos.**
- **Cantidad de defunciones infantiles.**

En la sección 3.2.2, se explicará como se han generado concretamente estos datos.

3.2. Backend

Antes de comentar las diferentes características implementadas en el Backend, voy a nombrar las tecnologías que vamos a emplear:

- **Node.js:** Es el framework principal del Backend, y gracias al cual instalaremos todas las librerías y dependencias.
- **Express.js [46]:** Permite montar una API REST sencilla, para poder comunicar al servidor con la base de datos o Backend.
- **Mongoose:** Es una herramienta enfocada en su uso junto a MongoDB o MongoDB Atlas. Permite fácilmente, crear, borrar o alterar datos en la base de datos.
- **Node-cron [47]:** Simula el comportamiento de un cron de Linux.
- **Axios [48]:** Herramienta que facilita la manera de realizar peticiones HTTP.
- **PM2 [49]:** Permite ejecutar en segundo plano, varios ficheros Javascript a la vez. Se hablará de ella en detalle en la sección de despliegue.
- **Morgan [50]:** Mostrará un log de las peticiones que se realicen a la API de Express. Esta librería solo se va a emplear durante la fase de desarrollo.

3.2.1. Obtención de datos

Para poder tratar los datos, previamente debemos de conseguirlos. La manera que vamos a emplear para obtenerlos, es usando **Axios** para realizar peticiones **GET** a la API.

Lo primero que debemos de hacer es una petición GET al dataset que queremos obtener, para ello necesitamos añadir a la **URL base** de la API, el **ID** del dataset. La URL base es la siguiente:

```
https://datos.gob.es/apidata/catalog/dataset/
```

Si la petición ha sido exitosa, no obtendremos directamente los campos del dataset. Por su parte obtendremos un listado de enlaces, ya que el mismo dataset se puede obtener en distintos formatos: xml, json, csv, etc.

Lo que requiere hacer ahora, es una petición al enlace que indique que está en formato json, ya que es con el vamos a trabajar. Una vez realizada esta segunda petición, ya tendremos nuestro dataset listo para tratar.

3.2.2. Tratamiento de datos

Cada dataset está realizado de una forma distinta y no siguen ningún patrón entre ellos, por lo que para cada uno de estos datos, debemos de realizar una implementación diferente.

Después de tratar los conjuntos de datos obtenidos de la API, hemos recopilado la siguiente información:

- **Población:** según la comunidad autónoma, el año y el género.

- **Índice de natalidad:** según la comunidad autónoma y el año.
- **Índice de fecundidad:** según la comunidad autónoma y el año.
- **Tasa de mortalidad infantil:** según el año y género.

Sin embargo, como mencioné en la sección 3.1, a partir de estos datasets hemos generado nuevos datos. A continuación veremos como se han conseguido estos resultados:

- **Cantidad de nacimientos:** Se ha generado mezclando los datos de “Población” e “Índice de natalidad”. Para obtener la cantidad de nacimientos, se ha empleado la fórmula del índice de natalidad, la cual es de la siguiente manera:

$$indiceNatalidad = \frac{nacimientos}{poblacion} * 1000 \quad (3.1)$$

Si observamos detenidamente la fórmula, vemos que el único dato que no poseemos es “nacimientos”, que es justo lo que estamos buscando. Entonces si despejamos los valores, nos queda la fórmula a continuación :

$$nacimientos = \frac{indiceNatalidad * poblacion}{1000} \quad (3.2)$$

Ahora, si aplicamos dicha fórmula con los datos que tenemos, finalmente obtendríamos la *cantidad de nacimientos* según el año y la comunidad autónoma.

- **Cantidad de defunciones infantiles:** Se ha generado mezclando el nuevo dataset creado “Cantidad de nacimientos”, con “Tasa de mortalidad infantil”. En este caso, la fórmula usada ha sido la de la tasa de mortalidad infantil:

$$tasaMortalidadInfantil = \frac{nMenoresFallecidos}{nacimientos} * 1000 \quad (3.3)$$

Donde “nMenoresFallecidos” hace referencia a los menores de un año fallecidos. Si despejamos justamente este valor, la fórmula quedaría así:

$$nMenoresFallecidos = tasaMortalidadInfantil * nacimientos * 1000 \quad (3.4)$$

Por lo tanto, si empleamos esta fórmula con los datasets mencionados previamente, generaríamos la *cantidad de defunciones infantiles* según el año.

Finalmente, si contamos con estos dos datasets generados por nosotros, tendríamos un total de seis conjuntos de datos realizados para el prototipo.

3.2.3. Automatización de actualizaciones

Como se ha mencionado previamente, los datos se deben de actualizar automáticamente, para ello haremos uso de la herramienta **node-cron**. Se trabaja con una expresión base como esta: ‘*****’. Donde cada asterisco significa lo siguiente:

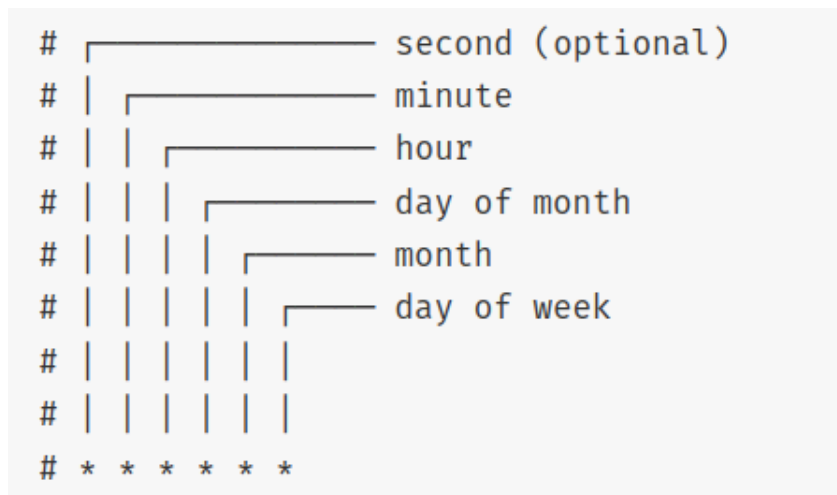


Figura 3.1: Tiempos de referencia en node-cron

Lo que continua por hacer ahora, es definir cada cuanto tiempo se debe ejecutar cada tarea. Esto lo averiguamos en la página web del repositorio de datos abiertos del Gobierno de España, donde indica la frecuencia de actualización y cuando fue la última actualización. Si por ejemplo la frecuencia es anual y se realiza cada día 1 de enero, la expresión debería ser la siguiente: `'* * * 1 1 *'`.

Si aplicamos esto a cada conjunto de datos que poseemos, las expresiones quedan de la siguiente manera:

- **Población:** `0 0 12 * * 4`. Cada jueves a las 12:00.
- **Índice de natalidad:** `0 0 12 9 12 *`. Cada 9 de diciembre a las 12:00.
- **Índice de fecundidad:** `0 0 12 9 2 *`. Cada 9 de febrero a las 12:00.
- **Tasa de mortalidad infantil:** `0 0 12 9 12 *`. Cada 9 de diciembre a las 12:00.
- **Cantidad de nacimientos:** `0 10 12 9 12 *`. Cada 9 de diciembre a las 12:10.
- **Cantidad de defunciones infantiles:** `0 20 12 9 12 *`. Cada 9 de diciembre a las 12:20.

Lo que restaría para finalizar, sería indicar qué métodos o secuencias son las que deben repetirse, esto lo conseguimos indicando en el código lo siguiente:

Listado 3.1: Ejemplo de uso de node-cron

```

import cron from 'node-cron';
cron.schedule({{cron-expression}}, async () => {
  // Código a repetir
});

```

Donde `"{{cron-expression}}"` significa el periodo de tiempo en el que debe repetirse.

3.2.4. Base de datos

La base de datos que vamos a emplear es **MongoDB Atlas** junto con la herramienta de mongoose, tal y como mencioné previamente.

Al tratarse de una base de datos no relacional, en vez de tablas contamos con colecciones, y en lugar de filas de una tabla, tenemos documentos de una colección. Teniendo esto en cuenta, se han creado seis colecciones, una para cada dataset. Y dentro de cada colección, la cantidad de documentos varía en función de la cantidad de años, o de comunidades autónomas disponibles.

Collection Name	Documents
amount_of_births	19
fecundity	19
infant_defunctions_per_year	19
infant_mortality_rate	46
nativity	19
population	19

Figura 3.2: Nombre de las colecciones en la base de datos en MongoDB Atlas

Dataset	Nº de documentos	Entradas
Población española	19	Una por comunidad autónoma
Índice de natalidad	19	Una por comunidad autónoma
Índice de fecundidad	19	Una por comunidad autónoma
Tasa de mortalidad infantil	46	Una por año (2020-1975)
Cantidad de nacimientos	19	Una por comunidad autónoma
Índice de fecundidad	19	Una por comunidad autónoma

Tabla 3.2: Especificación del número de entradas por cada dataset

Para crear las colecciones desde el Backend, debemos de realizar un modelo y esquema (o schema) por cada conjunto de datos. En estos esquemas indicamos la estructura que queremos que siga la colección.

Listado 3.2: Ejemplo de esquema y modelo con mongoose

```
import mongoose from 'mongoose';
const ExampleSchema = new mongoose.Schema({
  value: {type: Number, required: true},
  name: {type: String, required: true}
```

```
});
```

```
const ExampleModel = mongoose.model('Example', ExampleSchema);
```

Finalmente, ya con los modelos creados podemos realizar operaciones CRUD a la base de datos.

3.2.5. API REST

Para evitar que los usuarios puedan conectarse directamente a la base datos, crearemos una API REST mediante la herramienta **Express.js**.

La base de datos solamente va a ser modificada por el Backend, por lo que la única operación que tenemos que implementar es la petición *GET*. Para realizar dichas peticiones a MongoDB Atlas, emplearemos los modelos creados en la sección previa (3.2.4).

Listado 3.3: Ejemplo de petición GET a MongoDB Atlas usando Express y mongoose

```
const ExampleRouter = express.Router();

ExampleRouter.get('/example', async (req, res) => {
  const example = await ExampleModel.find();
  res.send(example);
});
```

Como se puede apreciar en el ejemplo previo, debemos de definir los **endpoints**¹ con los que trabajaremos. Es por ello, que por cada modelo/dataset, crearemos un método que implemente una petición *GET*, a un endpoint específico.

El nombre asignado a las rutas fue el siguiente:

Dataset	Endpoint
Población española	/poblacion
Índice de natalidad	/natalidad
Índice de fecundidad	/fecundidad
Tasa de mortalidad infantil	/tasa-de-mortalidad-infantil
Cantidad de nacimientos	/nacimientos
Índice de fecundidad	/defunciones-infantiles

Tabla 3.3: Endpoints elegidos para cada dataset

3.3. Frontend

Como se mencionó en la sección 2.2.2 del estudio previo, el framework que vamos a emplear es Ionic en conjunto a Vue, pero a su vez también usaremos las siguientes tecnologías: **TypeScript**, **Vue-Charts**, **Axios** y **CSS**.

¹**Endpoint:** También conocido como punto final. Es el extremo de una conexión con la API (ruta), donde se recibe una llamada.

Llevar a cabo el prototipo será prácticamente como realizarlo para Vue, pero añadiendo algunas etiquetas y características propias de Ionic.

3.3.1. Llamadas a la API

Para obtener los distintos datasets en el Frontend, debemos de realizar llamadas a la API que montamos previamente en el Backend.

La manera realizada para conseguir ejecutar estas llamadas, fue crear dentro de una carpeta denominada “**services**”, una instancia de Axios y un servicio por cada dataset. De esta manera, se usaba Axios en cada servicio, para llamar a los distintos endpoints definidos y obtener los datos.

Listado 3.4: Estructura de la carpeta "services"

```
services/  
|— amountOfBirthsService.ts  
|— axiosInstance.ts  
|— fecundityService.ts  
|— infantDefunctionsService.ts  
|— infantMortalityRateService.ts  
|— natalityService.ts  
|— populationService.ts
```

Con esto ya tenemos los distintos métodos para obtener la información, sin embargo, ahora necesitamos saber dónde y cuándo usarlos. La manera más óptima sería realizar las peticiones antes de que se cargue la página. Esto se puede hacer fácilmente con el método *onBeforeMount()* que proporciona Vue.

Listado 3.5: Ejemplo de llamada a “onBeforeMount”

```
setup() {  
  ...  
  ...  
  onBeforeMount(async () => {  
    fetchData();  
  });  
}
```

Este método lo que hace concretamente, es ejecutar todo su contenido justo antes de que se monte el componente o página.

3.3.2. Generación de gráficos

Una vez obtenido todos los datos, lo que resta es mostrarlos. La idea es que el usuario pueda cambiar de gráfica, género, año, etc., de manera dinámica.

Para lograrlo, en lo primero en lo que vamos a enfocarnos, es en obtener las distintas etiquetas² posibles para cada dataset. Dentro de cada servicio, crearemos los métodos necesarios para conseguir las etiquetas que necesitemos, los cuales son:

²**Etiquetas:** Engloba a los títulos o leyendas que se ponen en el eje X de la gráfica

- **getYearsAsLabels():** Muestra los distintos años disponibles como etiqueta en las gráficas.
- **getAutonomousCommunitiesAsLabels():** Muestra las distintas comunidades como etiqueta en las gráficas.

Ahora se debe de filtrar los valores que el usuario quiere que se muestre en el **eje Y**, para ello también se han creado varios métodos en cada servicio en función de las necesidades:

- Mostrar los valores en la distintas comunidades autónomas, según un género o el total.
- Mostrar los valores en la distintas comunidades autónomas, según la combinación de uno o varios géneros (total incluido).
- Mostrar los valores durante distintos años en una comunidad autónoma, según un género o el total.
- Mostrar los valores durante distintos años en una comunidad autónoma, según la combinación de uno o varios géneros (total incluido).

Ya con todo lo realizado hasta el momento, tenemos lo necesario para realizar las gráficas. Para ello, creamos un componte de Vue-Charts por cada tipo de gráfica que queramos realizar. En este caso, vamos a realizar 6 tipos de gráficas distintas: **barras, líneas, radar, área polar, circular y donut**. Estos componentes recibirán las etiquetas y los valores a mostrar en la gráfica.

En el apartado 3.3.4 se mostrará como se ven las distintas gráficas.

3.3.3. Conversión a Android

Se ha mencionado con anterioridad, que el prototipo debe estar disponible tanto en versión web, como en **.apk**³ para dispositivos Android.

Cuando se realiza el prototipo de Ionic, se está trabajando en la versión web. Para convertirlo a versión Android, sólomente hay que realizar varios comandos y hacer algunas configuraciones como indica el sitio oficial de Ionic [51]. Aparte, también necesitaremos generar el logo que tendrá nuestra aplicación, para ello deberemos seguir otras instrucciones proporcionadas por Capacitor [52].



Figura 3.3: Imágenes usadas para el logo de la aplicación

³.apk: Es la extensión que se usa en los paquetes o extensiones de android

Una vez seguidos los pasos del artículo, tendremos un proyecto creado en **Android Studio** [53]. Finalmente, si dentro del proyecto, en la barra de herramientas accedemos a **Build** → **Build Bundle(s)/APK(s)** → **Build APK(s)**, obtendremos nuestra *APK de visualización de datos demográficos en España* [54].

Cabe destacar que con Ionic también se puede generar la aplicación para iOS en formato **.ipa**⁴. Sin embargo, es necesario hacer uso de **Xcode** [55], aplicación exclusiva de dispositivos MAC y algo de lo que actualmente carezco.



Figura 3.4: Logos de entornos de trabajo para el desarrollo móvil (2)

3.3.4. Pantallas

El prototipo se ha desarrollado de manera *responsiva* tanto para web como para móvil, ya que es de gran importancia que las gráficas se puedan visualizar correctamente desde cualquier dispositivo. Es por ello, que entre las distintas versiones se encuentran algunas diferencias.

Existen 2 tipos de pantallas:

- **Inicio:** Destinado solamente para la página de inicio.
- **Visualización:** Empleado por las distintas colecciones de datos que poseemos.

En esta sección se mostrarán las pantallas de la versión web, y se expondrán las diferencias realizadas de la versión móvil.

Versión Web

- **Inicio:** Esta es la página principal, aquí podemos encontrar un resumen acerca del funcionamiento y objetivo de la aplicación. A su vez también encontramos una serie de paneles con una breve descripción cada uno, estos hacen referencias a las distintas colecciones de datos, si hacemos clic sobre algún panel, nos llevará a la página de ese dataset.

⁴.**ipa**: Es la extensión que se usa en los paquetes o extensiones de iOS

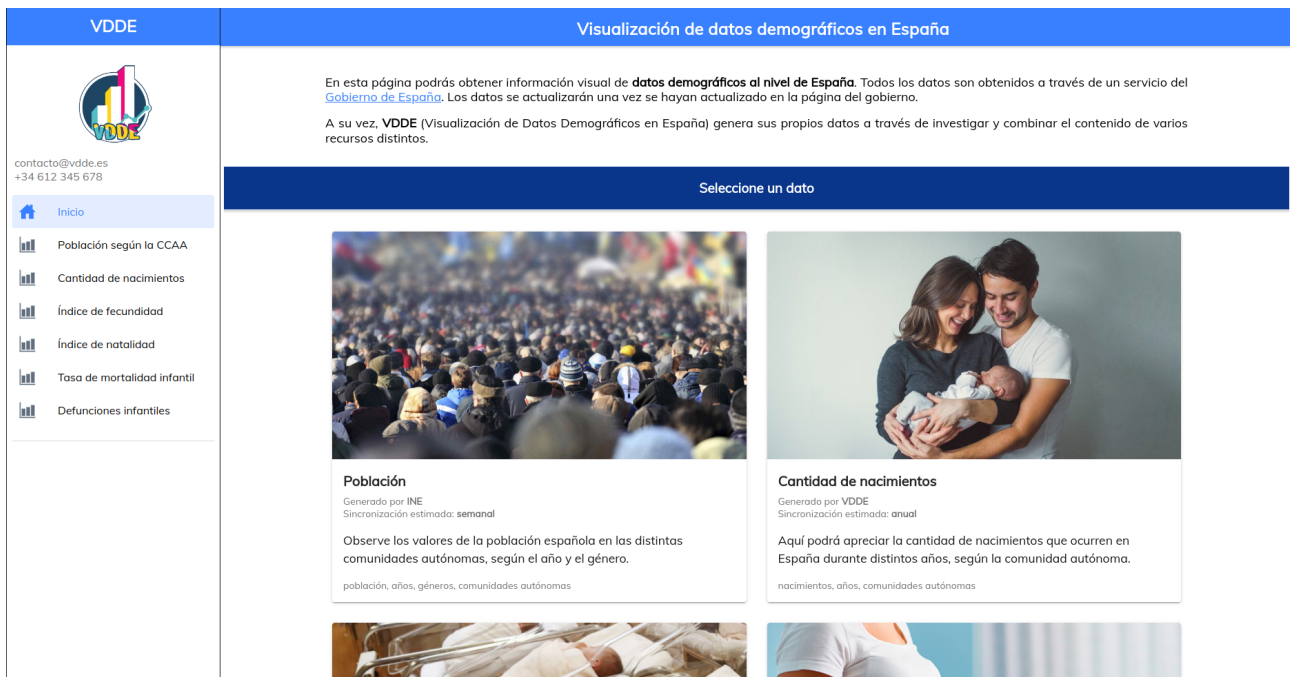


Figura 3.5: Página de inicio

- Menú:** Se encuentra en todas las páginas en la parte izquierda, es un menú que cuenta con: el logo de la aplicación, información de contacto, y accesos directos a todas las pantallas del sitio. A su vez, se muestra marcado en azul, la página en la que nos encontramos actualmente. En el resto de capturas se ignorará este menú, ya que es el mismo siempre.



Figura 3.6: Menú desplegable

- **Visualizaciones:** En el prototipo existen 6 colecciones de datos, por lo que hay 6 pantallas de visualizaciones. En estas pantallas podremos personalizar los valores o gráficas que se muestran entre las opciones.

Como dentro de cada página se pueden realizar bastantes modificaciones a las gráficas, sólo se va a mostrar un dataset en detalle con varios ejemplos, si quiere ver el resto de datos y probarlos, puede acceder al sitio web: <https://www.vdde.me> [56]. El dataset que se va a emplear es **Población según la comunidad autónoma**.

Población según la comunidad autónoma

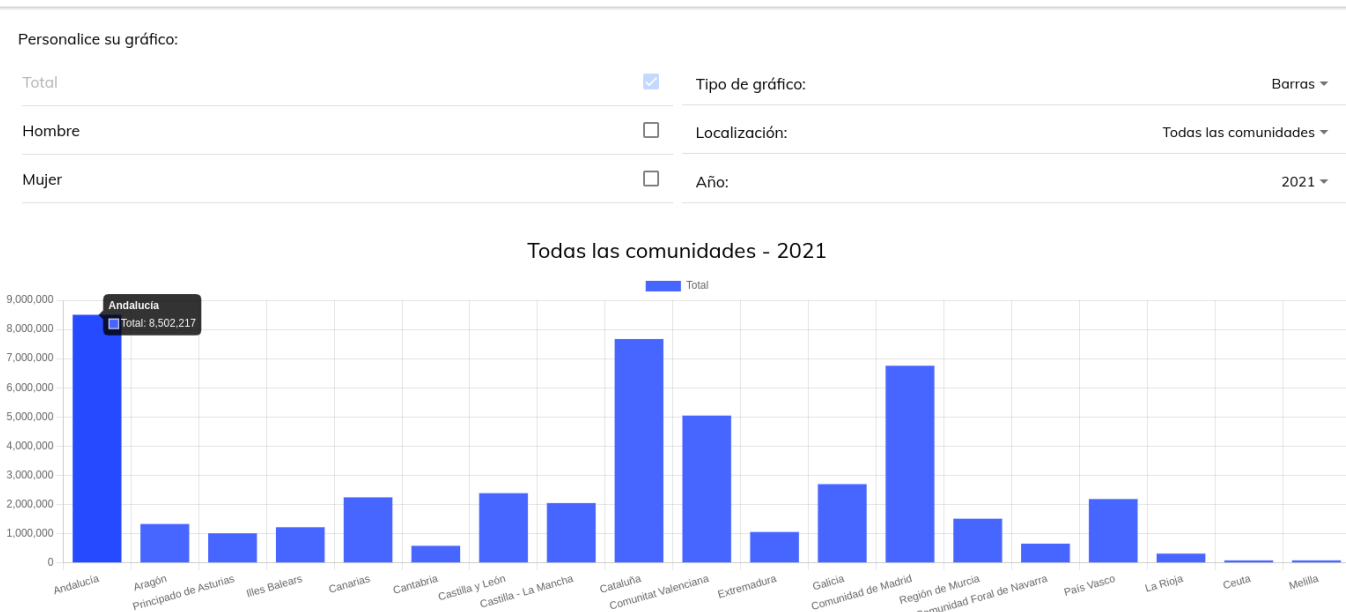


Figura 3.7: Diagrama de barras sobre el total de la población española en 2021



Figura 3.8: Gráfico de líneas de la población en 2021 sobre el total, hombres y mujeres

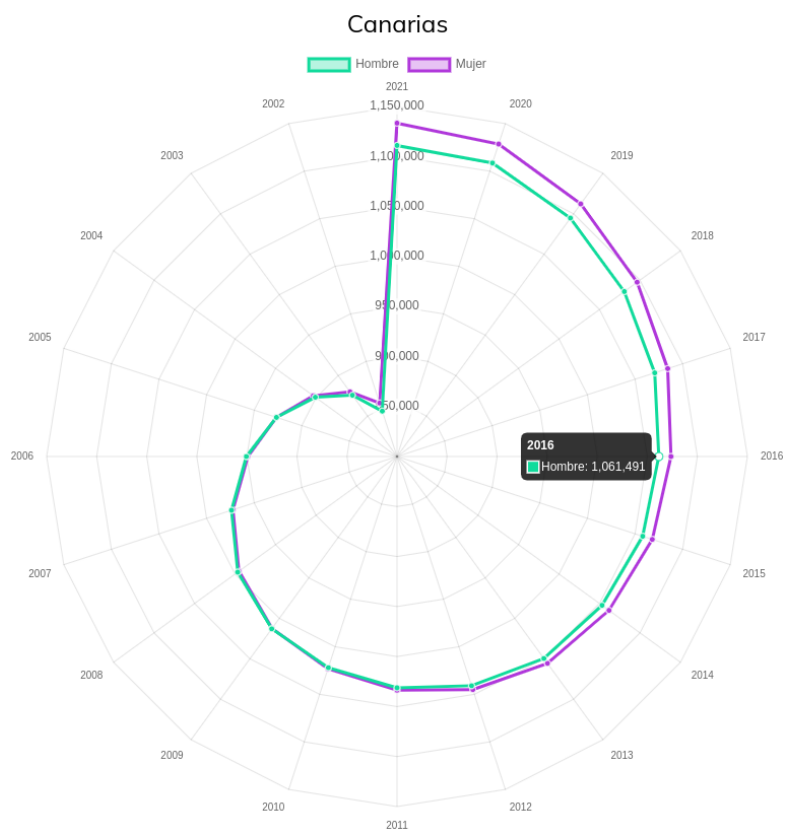


Figura 3.9: Gráfico de radar de la población en Canarias sobre hombres y mujeres, durante distintos años

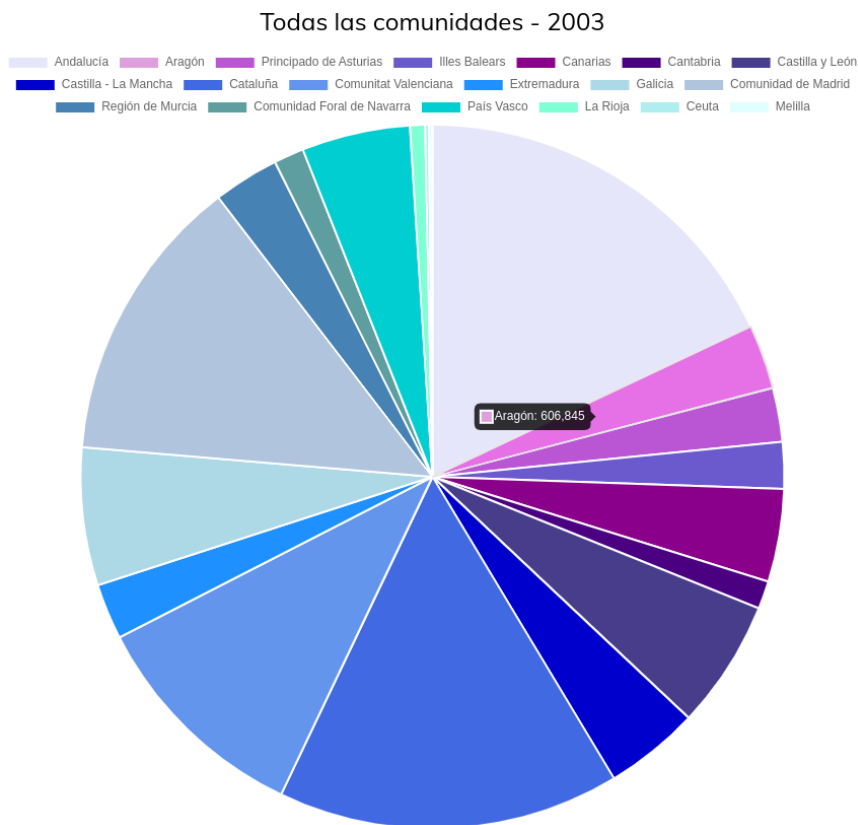


Figura 3.10: Gráfico circular de la población de hombres en 2003

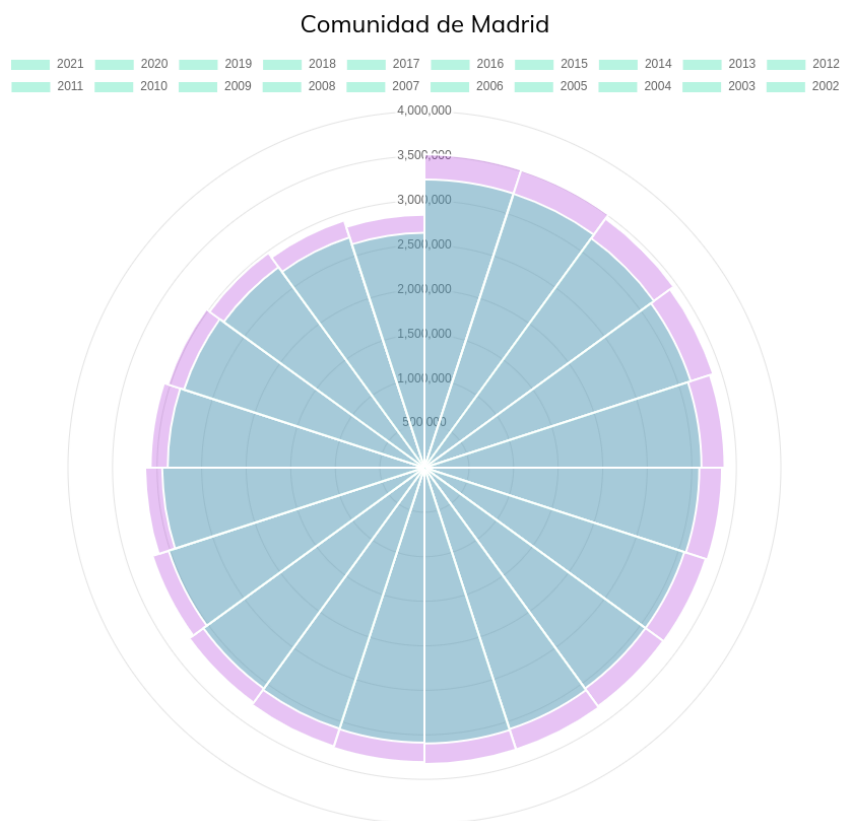


Figura 3.11: Gráfico de área polar de la población de hombres y mujeres en Madrid, durante distintos años

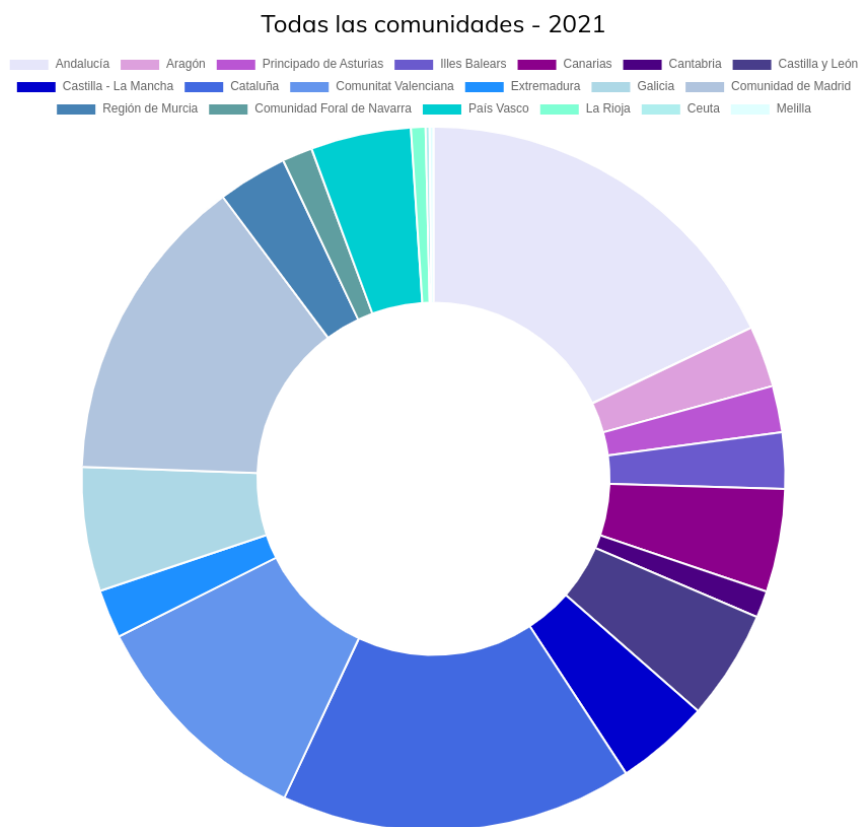


Figura 3.12: Gráfico donut de la población total en 2021

Información

En esta página podrá apreciar los valores de la población española en las distintas comunidades autónomas, según el año y el género.

- Generado por: [INE](#)
- Frecuencia de actualización: Semestral
- [Enlace al sitio](#)
- [Accede al dataset](#)

Guía de uso

- Se puede elegir un género o varios a la vez.
- Para las gráficas "Circular" y "Donut", sólo se puede elegir un género.
- Para el resto de gráficas si se pueden elegir varios géneros si está disponible la opción.
- Cuando la opción "Todas las comunidades" está seleccionada, se puede elegir un año.
- Cuando se especifica una comunidad autónoma, aparecen todos los valores en los distintos años para esa comunidad autónoma.
- Si no se puede especificar la localización, significa que se muestran todas las comunidades.
- Si no se puede especificar un género, significa que se está mostrando el total de la población.
- Si pasa el cursor o pulsa sobre algún elemento de la gráfica, podrá ver su valor total.

Figura 3.13: Información del dataset

Versión Móvil

La versión de móvil es prácticamente idéntica a la versión web, a continuación se comentarán las diferencias. El **menú** es igual al sitio web, sin embargo, ahora se encuentra escondido en un menú desplegable. Por otra parte, en la **pantalla de inicio**, solamente se dispone de un panel por fila en vez de dos.



Figura 3.14: Menú desplegable oculto y pantalla de inicio

En cuanto a las **visualizaciones**, de manera vertical se ven correctamente los valores. No obstante, usar la aplicación de manera horizontal es obviamente mejor debido al aprovechamiento de la pantalla.



Figura 3.15: Gráfico de líneas en versión móvil vertical

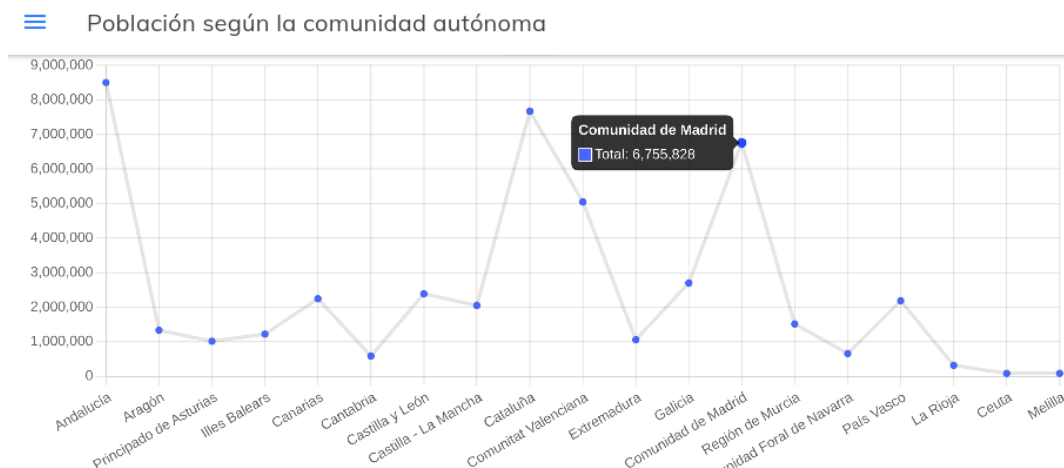


Figura 3.16: Gráfico de líneas en versión móvil horizontal

3.4. Despliegue

Actualmente todo lo desarrollado se encuentra en una máquina local, por lo que no es accesible para el resto de personas. En esta sección desplegaremos tanto el Backend como el Frontend en DigitalOcean.

3.4.1. Creación y configuración de la máquina

El primer paso para realizar el despliegue, es crear un *droplet*. En concreto, las características de nuestro droplet son: una CPU, un GB de RAM, 25GB de almacenamiento SSD y un sistema operativo Linux.

Lo siguiente realizado, fue obtener el dominio “**vdde.me**” en **Namecheap**. Dicho dominio hay que vincularlo en las opciones de DNS tanto de DigitalOcean como de Namecheap. En total contamos con la siguiente configuración:

- **www.vdde.me** y **vdde.me**: Acceso a la página web.
- **www.api.vdde.me** y **api.vdde.me**: Acceso a la API montada en el Backend.

Para que nos redireccione correctamente a las páginas que deseamos, tenemos que hacer uso de **Nginx**⁵. En concreto, tenemos que indicar que los dominios **www.api.vdde.me** y **api.vdde.me** van a escuchar en el puerto *3000*, que es donde se encuentra levantado el Backend y la API de Express.

Mientras que por otro lado indicamos que los dominios **www.vdde.me** y **vdde.me** van a escuchar en el puerto *80*, y además, definimos la ruta de donde se encuentra nuestro fichero **index.html** del prototipo.

Finalmente, tenemos que generar los certificados SSL de los subdominios que estamos empleando. Lo único que debemos de hacer para obtener esto, es indicar dichos subdominios a la herramienta de **Certbot** y ya obtendríamos los certificados.

3.4.2. Backend

Para mantener el Backend en constante funcionamiento, debemos de emplear **PM2**. Lo que va a hacer esta herramienta, es crearnos un proceso “demonio”⁶ del archivo que queramos. De esta manera, la API creada, y el cron para actualizar los datos, estarán siempre en funcionamiento.

3.4.3. Frontend

El proyecto en Ionic con Vue no se puede desplegar directamente, necesitamos generar los archivos estáticos del proyecto. Para ello simplemente debemos realizar el comando `npm run build`, y ya tendríamos el proyecto listo para el despliegue.

⁵**Nginx**: herramienta que actúa como proxy inverso, es decir redirecciona las solicitudes recibidas a puertos concretos, que ejecutan ciertas acciones

⁶**Proceso demonio**: proceso que se ejecuta en segundo plano

Capítulo 4

Estudio de viabilidad económica

En esta sección se planteará un desarrollo del prototipo desde un punto de vista profesional. La finalidad de esto es identificar si las características que tendrá nuestro producto, son lo suficientemente potentes y viables, como para rivalizar contra la competencia del sector.

4.1. Funcionalidades y herramientas

Las características que tiene el prototipo actual, no serán todas las propiedades que tendrá la versión final. Aparte de facilitar, actualizar y visualizar datos, como contaremos con mayor tiempo y presupuesto, algunas de las funcionalidades que se añadirán a la versión final son las siguientes:

- **Registro e inicio de sesión:** actualmente no existe un control de quienes son los usuarios que entran a nuestra aplicación. Además, para incluir los distintos planes que tendrá la aplicación, es estrictamente necesario el uso de un inicio de sesión.
- **Creación de datos:** los usuarios podrán generar su propia información a partir de las funcionalidades proporcionadas por la aplicación.
- **Favoritos y guardar estado de edición:** poder guardar aquellos datos que más se visiten por un usuario, dentro de su sección de favoritos. A su vez, también se implementará un sistema donde, si el cliente ha estado editando un dataset, cuando vuelva a la página se cargará de nuevo ese estado.
- **Más tipos de gráficas y combinaciones:** se proporcionarán más variedad de gráficas (como mapas, burbuja, diagrama de caja y bigotes...), y poder usar la combinación de algunos de ellos para una mejor visualización.
- **Marketplace:** se implementará una zona de mercado, donde los clientes podrán subir contenido que pueda ser relevante para el resto de usuarios dentro de la aplicación, como por ejemplo, algún diseño que haga las gráficas más llamativas.
- **Modelo predictivo de datos:** existirá un algoritmo que podrá ser entrenado con distintos valores de uno o varios conjuntos de datos, a fin de predecir o estimar futuros resultados.

- **Mejores filtros:** los usuarios podrán personalizar aún más los datasets de la aplicación y los creados por ellos mismos.
- **Mayor cantidad de valores estadísticos:** se proporcionará más información a los usuarios, con valores como: correlación de datos, media, mediana, moda, mínimos, máximos...
- **Búsqueda avanzada:** el prototipo carece de un buscador de conjunto de datos, ya que solamente existen 6. Sin embargo, la versión final necesita un buscador para explorar de mejor manera todos los datos existentes, a su vez, contará con un filtro para poder realizar búsquedas con mayor precisión.
- **Sección de noticias:** contará con las últimas novedades añadidas a la aplicación, así como los últimos acontecimientos en el mundo de la demografía.

4.2. Desarrollo del proyecto

Ahora veremos las diferentes fases y tareas que se deberían llevar a cabo para realizar este proyecto, así como la cantidad de recursos, costes y duración. Para ello, vamos a emplear la herramienta de gestión y administración, **ProjectLibre**¹.

A continuación definiremos las distintas fases del proyecto:

- **Fase de análisis:** Esta primera fase comprende responder una serie de dudas planteadas al inicio, tales como: ¿Qué es lo que queremos obtener con el proyecto?, ¿Cómo lo vamos a obtener?, ¿Cómo se encuentra la competencia en el sector y cómo se gestionan? Tras realizar el estudio de todas estas cuestiones, estaríamos listos para avanzar a la siguiente fase.
- **Fase de diseño:** En esta segunda etapa se definirán: las distintas herramientas o tecnologías a emplear, la arquitectura de la base de datos, los módulos del Backend y Frontend, los datasets a tratar, los diseños de las versiones web y móvil, casos de uso, y el nombre y logo del proyecto.
- **Fase de desarrollo:** Es la fase más larga, y aquí ya disponemos de todo lo necesario para empezar el desarrollo de la aplicación. Se llevan acciones como:
 - Implementación de la base de datos.
 - Desarrollo del Backend: algoritmo para el procesamiento de datos, API REST, automatizaciones, testing...
 - Desarrollo del Frontend: hojas de estilos, tipografía, conexión con la API REST, creación de las distintas páginas, los métodos para filtrar datos, las gráficas, testing...
 - Conversión a iOS y Android.
 - Revisión del proyecto.

¹Véase: <https://es.wikipedia.org/wiki/ProjectLibre>

- **Fase de despliegue:** Ahora entramos en la última fase. Lo que haremos aquí será lanzar nuestro producto en las distintas plataformas para comenzar con su comercialización.

El trabajo será telemático, aprovechando las ventajas que ofrece tanto a los trabajadores como a la empresa. No obstante, es necesario algunos materiales para el desarrollo:

- Ordenador de trabajo de Apple (1200€): para generar la aplicación en dispositivos iOS, es necesario emplear un dispositivo MAC.
- Móvil iOS para las pruebas (350€).
- Móvil Android para las pruebas (250€).
- Licencia Play Store (25€).
- Licencia App Store (100€/año).
- Dominio en Namecheap (21€/año).
- Certificado SSL (18€/año).
- Servidor DigitalOcean (84€/mes): 16GB de RAM, 2 CPUs, 50GB SSD, 4TB de transferencia. Aquí se encontrarán alojados el Backend, el Frontend, y la base de datos de MongoDB.

Ya con las tareas y recursos definidos, necesitamos identificar a los trabajadores del proyecto:

- Un **jefe de proyectos (40€/h)**: persona encargada de supervisar el avance y tareas realizadas del proyecto. A su vez, es quien controla y administra los recursos.
- Un **analista de software (28€/h)**: interviene en la toma de decisiones para especificar las necesidades del proyecto, y como se deben cubrir para alcanzar el objetivo.
- Un **arquitecto de software (30€/h)**: tiene el conocimiento adecuado para establecer los requisitos, tecnologías, herramientas y estructuras, del proyecto.
- Un **diseñador gráfico (20€/h)**: es la persona responsable de diseñar el logo de la aplicación, así como las distintas pantallas en las versiones web y móvil.
- Un **tester (21€/h)**: durante el desarrollo del proyecto, a las distintas funcionalidades que se vayan creando, el tester comprobará el correcto funcionamiento de estas características.
- Un **quality assurance (23€/h)**: es la persona que evaluará tanto la calidad como la sostenibilidad del código, para que cumpla con los requerimientos establecidos.
- Cuatro **programadores (30€/h cada uno)**: dos programadores estarán enfocados en el Backend del proyecto, y en algunas configuraciones y despliegues en el servidor. Los otros dos trabajadores realizarán el Frontend y las conversiones a móvil.

Ahora, si añadimos toda esta información a **ProjectLibre**, con sus respectivos tiempos, nos queda que la duración estimada del proyecto son **191 días laborables**, con una jornada de *8 horas de lunes a viernes*. El periodo abarcaría desde la fecha **01/03/2022**, hasta el **25/11/22**, contando con festivos y fines de semana. Finalmente, el desarrollo total del proyecto está estimado en **126.804,00€**.

No obstante, después del despliegue se hará un mantenimiento mensual con un coste de 300€.

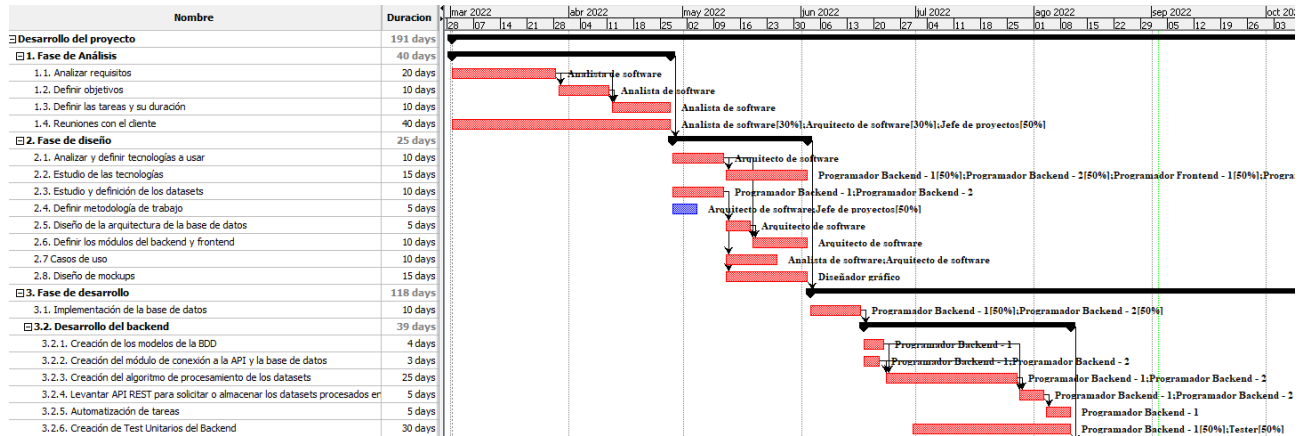


Figura 4.1: Diagrama de Gantt (1)

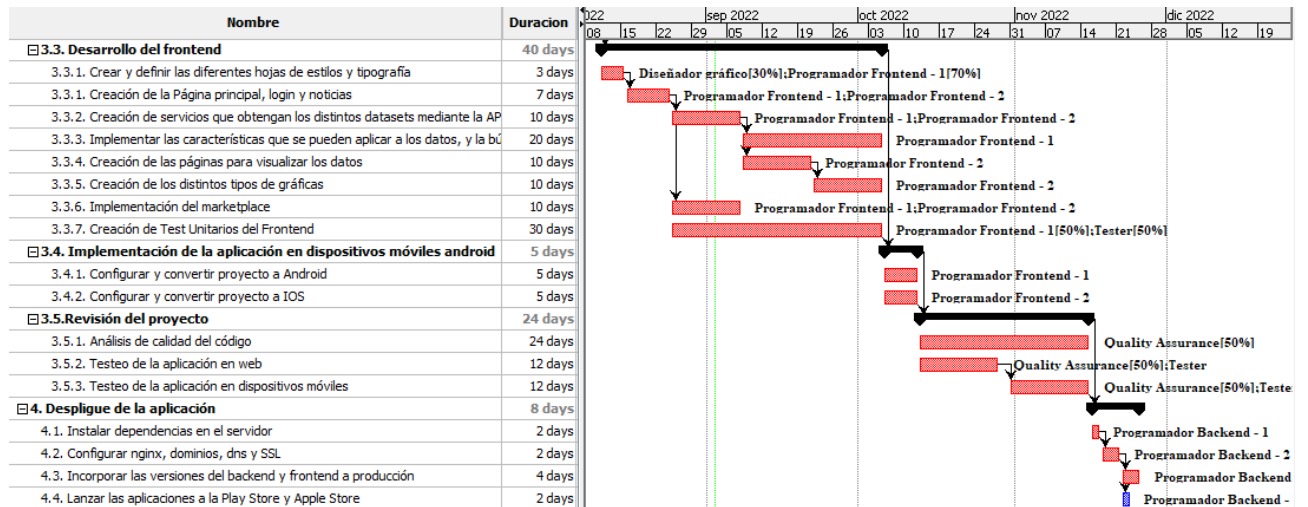


Figura 4.2: Diagrama de Gantt (2)

4.3. Comercialización

4.3.1. Fuentes de ingreso

A continuación, nos encontramos con la problemática de cómo obtendríamos el beneficio de nuestro producto. Para responder a esto, se han establecido una serie de planes:

- **Freemium:** como parte del objetivo de nuestro proyecto es que la gente sea consciente de la situación demográfica del país, la versión **Freemium** permite a los

usuarios *visualizar gratuitamente* los datos más relevantes del momento en la web. De esta manera, aquellas personas que solamente se quieran informar de los datos actuales, no se ven en la necesidad de pagar una suscripción.

Sin embargo, para obtener beneficios con este plan, se empleará **Google AdSense** [57]. También, por cada dato que se quiera observar que no sea de la versión *Freemium*, el usuario podrá acceder pero después de ver un anuncio publicitario.

- **Visualizer (5€/mes):** este plan está destinado a aquel sector de la población que sólo busca informarse y adquirir conocimiento de la demografía del país. Con esta versión podrás visualizar todos los datos disponibles desde la web, y también con la aplicación para móviles.
- **Curiosity (10€/mes):** todas las características de **Visualizer** se encuentran disponibles en **Curiosity**. Además de esto, permite a los usuarios poder editar los datasets disponibles, a fin de obtener otros resultados.
- **Premium (18€/mes):** incluye todas las características de **Curiosity**, pero además puedes crear tu propia información y gráficas desde cero. También puedes trabajar en una misma sesión con otros compañeros en simultáneo, emplear características en fase beta, y cuenta con un servicio de atención al cliente.

Aparte de los distintos planes, también existen otras maneras de las que es posible que se pueda obtener beneficios o popularidad:

- **Mes de prueba:** con esta oferta, dejamos que el cliente obtenga gratuitamente las características premium durante un mes. De esta manera, permitimos que el usuario compruebe por sí mismo, si quiere nuestro producto.
- **Colaboraciones:** ofrecemos un periodo de pruebas mayor a ciertas instituciones o iniciativas, como Github Students o Universidades. Esto sería a cambio de publicitar nuestro producto y obtener beneficio económico.
- **Subvenciones del Gobierno:** ya que se trata de un proyecto que también busca que haya menos desinformación entre los ciudadanos, es posible que se puedan obtener ayudas con el objetivo de seguir fomentando la situación demográfica del país.

4.3.2. Estudio del retorno de inversión (ROI)

Con toda esta información, tenemos lo necesario para simular las visitas y ventas que tendrá nuestro producto, en función de la popularidad e interés de los temas tratados. Esto con el objetivo de averiguar si este proyecto es viable o no.

Durante la simulación se ha tenido en cuenta la facturación de: las licencias, servidor y mantenimiento del producto. También se ha estimado cuantas personas accederían a la web, y que tipo de plan comprarían.

Gasto acumulado y ROI

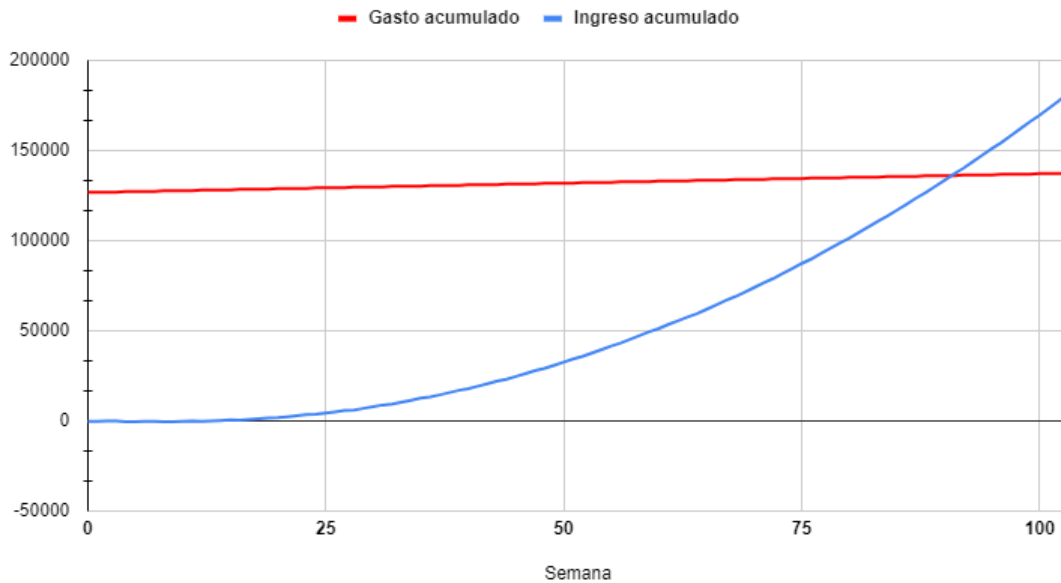


Figura 4.3: Gráfica que muestra el punto de retorno de inversión

Es por ello, que tras realizar un estudio² [58], se observó que las primeras semanas el proyecto se encontraba en números negativos, ya que el mantenimiento era superior a los beneficios generados. Sin embargo, a medida que avanzaban las fechas, tanto las visitas como el número de planes comprados fueron aumentando, obteniendo el **retorno de inversión** en la semana **89**, es decir, en **20 meses y medio** hemos recuperado la inversión inicial del proyecto junto con su mantenimiento.

Cabe destacar que para el estudio no se tuvo en cuenta el dinero que se podía haber obtenido a través de: subvenciones, colaboraciones, o algún tipo de publicidad externo a Google AdSense.

²Véase: <https://docs.google.com/spreadsheets/d/1qbtA0siYJnBTXglGsUc6skvqrj1RS9qfx2AB0dMs00c/edit?usp=sharing>

Capítulo 5

Conclusiones y líneas futuras

Aunque el prototipo contenga solamente una pequeña porción de información referente a la demografía en España, ha cumplido con la finalidad propuesta de la aplicación, la cual era generar datos que se actualicen automáticamente, para que los usuarios puedan estar informados de la demografía del país. También se ha cumplido con el resto de objetivos de la sección 1.4, ya que se ha demostrado la viabilidad de este proyecto.

Además, no solo sirve para que las personas sean conscientes de la situación actual, puede emplearse para otros casos como por ejemplo, estudiar acontecimientos pasados o estrategias de negocio enfocados a sectores de la población.

Como líneas futuras, aparte de lo mencionado en la sección 4.1, también se podrían enfocar los datos en temáticas relacionadas con la demografía, como podría ser la salud. De esta manera se podría expandir el área de conocimiento de la aplicación, y así atraer a más usuarios.

Centrándonos ahora en lo personal, con este proyecto, no solo he profundizado mis conocimientos acerca del Big data y la visualización de datos, sino que he podido aprender a usar tecnologías como; Vue, Ionic, Charts.js, Axios, etc., que hasta el momento, eran herramientas desconocidas para mi. A su vez, también he adquirido conocimientos acerca de como desplegar aplicaciones, generar certificados SSL, emplear dominios, y generar aplicaciones para dispositivos móviles.

Finalmente, pueden encontrar información de interés como el ProjectLibre realizado, o el estudio del ROI, en la carpeta compartida [54].

Capítulo 6

Summary and Conclusions

Although the prototype contains only a small portion of information about demography in Spain, it has fulfilled the proposed purpose of the application, which was to generate data that is updated automatically, so that users can be aware of the country's demography. The rest of the objectives of section 1.4, have also been met, since the viability of this project has been demonstrated.

In addition, it not only serves to make people aware of the current situation, it can be used for other cases, such as studying past events or business strategies focused on sectors of the population.

As future lines, apart from what is mentioned in section 4.1, the data could also be focused on topics related to demography, such as health. In this way, the area of knowledge of the application could be expanded, and in this way attract, more users.

Focusing now on my personal assessment, with this project, I have not only deepened my knowledge about Big data and data visualization, I have also been able to learn how to use technologies such as; Vue, Ionic, Charts.js, Axios, etc., which until now, were unknown tools for me. At the same time, I have also acquired knowledge about how to deploy applications, generate SSL certificates, use domains, and generate applications for mobile devices.

Finally, you can find information of interest such as the ProjectLibre carried out, or the ROI study, in the shared folder [54].

Capítulo 7

Presupuesto

Aparte del estudio de viabilidad económica del proyecto realizado, se ha llevado a cabo un presupuesto sobre el esfuerzo realizado durante el Trabajo de Fin de Grado. La duración de este se encuentra alrededor de unas **300 horas**, según lo establecido por el plan de estudios publicado el 21 de marzo de 2011, por la Universidad de La Laguna.

Tareas	Duración	Coste
Gestión y desempeño de un plan de negocios	30h	15€/h
Estudio previo de la competencia, y de las tecnologías disponibles para el desarrollo del prototipo	45h	15€/h
Desarrollo del Backend	50h	20€/h
Desarrollo del Frontend	100h	20€/h
Conversión a Android y Despliegue	15h	20€/h
Documentación	60h	15€/h
Total	300h	5.325€

Tabla 7.1: Presupuesto del Trabajo de Fin de Grado

Finalmente, este proyecto se ha llevado a cabo en **300h** de trabajo, con un coste de **5.325€**.

Bibliografía

- [1] “Consecuencias demográficas de la covid-19 en españa: entre la novedad excepcional y la reincidencia estructural,” Último acceso: 2022-08-26. [Online]. Available: <https://www.funcas.es/wp-content/uploads/2021/07/Esteve.pdf>
- [2] “Global big data industry market size 2011-2027 | statista,” Último acceso: 2022-08-26. [Online]. Available: <https://www.statista.com/statistics/254266/global-big-data-market-forecast/#:~:text=The%20global%20big%20data%20market,data%20market%20segment%20by%202027.>
- [3] “Total data volume worldwide 2010-2025 | statista,” Último acceso: 2022-08-26. [Online]. Available: <https://www.statista.com/statistics/871513/worldwide-data-created/>
- [4] “¿qué es el reto demográfico?” Último acceso: 2022-08-28. [Online]. Available: <https://www.miteco.gob.es/es/reto-demografico/temas/que-es/default.aspx>
- [5] “Investigadores, negacionistas, antivacunas y vendedores de humo... el quién es quién del coronavirus,” Último acceso: 2022-08-26. [Online]. Available: https://elpais.com/elpais/2020/09/04/planeta_futuro/1599206211_137960.html
- [6] “Expertos alertan del peligro de los bulos en las redes sociales para la futura vacunación,” Último acceso: 2022-08-26. [Online]. Available: <https://elpais.com/sociedad/2020-08-19/expertos-alertan-del-peligro-de-los-bulos-en-las-redes-sociales-para-la-futura-vacunacion.html#?rel=listaapoyo>
- [7] “Todos los tuits de miguel bosé, deconstruidos y explicados: ¿de qué habla exactamente?” Último acceso: 2022-08-26. [Online]. Available: https://elpais.com/elpais/2020/06/14/icon/1592128293_320734.html#?rel=listaapoyo
- [8] “Datos abiertos del gobierno de españa – datos.gob.es,” Último acceso: 2022-08-26. [Online]. Available: <https://datos.gob.es/es/>
- [9] “Datos abiertos del gobierno de españa – datos.gob.es,” Último acceso: 2022-08-26. [Online]. Available: <https://studylib.net/doc/8647594/3d-data-management--controlling-data-volume--velocity--an...>
- [10] “Estrategias y planes,” Último acceso: 2022-08-28. [Online]. Available: https://www.miteco.gob.es/es/reto-demografico/temas/Estrategias_Planes.aspx
- [11] “Publicada la convocatoria de ayudas por 16,2 millones para proyectos innovadores de transformación territorial y lucha contra la despoblación,” Último acceso: 2022-08-28. [Online]. Available: <https://www.miteco.gob.es/es/prensa/ultimas-noticias/Publicada-la-convocatoria-de-ayudas-por-16,>

2-millones-para-proyectos-innovadores-de-transformación-territorial-y-lucha-contra-la-despoblación
tcm:39-539894

- [12] “Visualización de datos | microsoft power bi,” Último acceso: 2022-08-20. [Online]. Available: <https://powerbi.microsoft.com/es-es/>
- [13] “Microsoft - sign up,” Último acceso: 2022-08-20. [Online]. Available: <https://signup.microsoft.com/get-started/signup?products=CFQ7TTC0L3PB>
- [14] “Comparación de productos y precios | microsoft power bi,” Último acceso: 2022-08-23. [Online]. Available: <https://powerbi.microsoft.com/es-es/pricing/>
- [15] “Casos de éxito de clientes | microsoft power bi,” Último acceso: 2022-08-23. [Online]. Available: <https://powerbi.microsoft.com/es-es/customer-showcase/>
- [16] “Cambiamos el modo de pensar en los datos,” Último acceso: 2022-08-20. [Online]. Available: <https://www.tableau.com/es-es/trial/tableau-software>
- [17] “Comprar tableau | tienda web de tableau,” Último acceso: 2022-08-20. [Online]. Available: <https://buy.tableau.com/es-es/>
- [18] “Nuestros clientes,” Último acceso: 2022-08-20. [Online]. Available: <https://www.tableau.com/es-es/our-customers>
- [19] “Spring boot,” Último acceso: 2022-08-23. [Online]. Available: <https://spring.io/projects/spring-boot>
- [20] “Node.js,” Último acceso: 2022-08-23. [Online]. Available: <https://nodejs.org/es/>
- [21] “Why quasar? | quasar framework,” Último acceso: 2022-08-23. [Online]. Available: <https://quasar.dev/introduction-to-quasar>
- [22] “Vue.js - the progressive javascript framework | vue.js,” Último acceso: 2022-08-23. [Online]. Available: <https://vuejs.org>
- [23] “Apache cordova,” Último acceso: 2022-08-23. [Online]. Available: <https://cordova.apache.org>
- [24] “Flutter - build apps for any screen,” Último acceso: 2022-08-23. [Online]. Available: <https://flutter.dev>
- [25] “Cross-platform mobile app development: Ionic framework,” Último acceso: 2022-08-23. [Online]. Available: <https://ionicframework.com>
- [26] “Capacitor: Cross-platform native runtime for web apps,” Último acceso: 2022-08-23. [Online]. Available: <https://capacitorjs.com>
- [27] “React – una biblioteca de javascript para construir interfaces de usuario,” Último acceso: 2022-08-23. [Online]. Available: <https://es.reactjs.org>
- [28] “React native · learn once, write anywhere,” Último acceso: 2022-08-23. [Online]. Available: <https://reactnative.dev>

- [29] M. Lynch, "Capacitor vs. cordova/phonegap: What is the difference? | ionic," Último acceso: 2022-08-06. [Online]. Available: <https://ionic.io/resources/articles/capacitor-vs-cordova-modern-hybrid-app-development>
- [30] C. Simmons, "Ionic vs. react native: Performance comparison," Marzo 2022, Último acceso: 2022-08-06. [Online]. Available: <https://ionicframework.com/blog/ionic-vs-react-native-performance-comparison/>
- [31] "Chart.js | open source html5 charts for your website," Último acceso: 2022-08-23. [Online]. Available: <https://www.chartjs.org>
- [32] "Home | vue-chartjs," Último acceso: 2022-08-23. [Online]. Available: <https://vue-chartjs.org>
- [33] "D3.js - data-driven documents," Último acceso: 2022-08-23. [Online]. Available: <https://d3js.org>
- [34] "Mongodb: La plataforma de datos para aplicaciones," Último acceso: 2022-08-23. [Online]. Available: <https://www.mongodb.com/es>
- [35] "Mongodb atlas database | multi-cloud database service," Último acceso: 2022-08-23. [Online]. Available: <https://www.mongodb.com/es/atlas/database>
- [36] "Firebase," Último acceso: 2022-08-23. [Online]. Available: <https://firebase.google.com/?hl=es>
- [37] "Mysql," Último acceso: 2022-08-23. [Online]. Available: <https://www.mysql.com>
- [38] "Mongoose odm v6.5.3," Último acceso: 2022-08-23. [Online]. Available: <https://mongoosejs.com>
- [39] "Cloud application platform | heroku," Último acceso: 2022-08-23. [Online]. Available: <https://www.heroku.com>
- [40] "Digitalocean | the developer cloud," Último acceso: 2022-08-23. [Online]. Available: <https://www.digitalocean.com>
- [41] "Buy a domain name - register cheap domain names from \$0.99 - namecheap," Último acceso: 2022-08-23. [Online]. Available: <https://www.namecheap.com>
- [42] "Name.com | domain names, registration, websites, hosting," Último acceso: 2022-08-23. [Online]. Available: <https://www.name.com>
- [43] "Certificado de clave pública - wikipedia, la enciclopedia libre," Febrero 2005, Último acceso: 2022-08-26. [Online]. Available: <https://ionicframework.com/blog/ionic-vs-react-native-performance-comparison/>
- [44] "Point to digitalocean nameservers from common domain registrars :: Digitalocean documentation," Octubre 2014, Último acceso: 2022-08-26. [Online]. Available: <https://docs.digitalocean.com/tutorials/dns-registrars/>
- [45] "Github - danielalvarezm/tfg-spain-demographic-data-visualization: Visualizador de datos demográficos en españa," Último acceso: 2022-09-04. [Online]. Available: <https://github.com/danielalvarezm/TFG-Spain-demographic-data-visualization>

- [46] "Express - node.js web application framework," Último acceso: 2022-08-23. [Online]. Available: <http://expressjs.com>
- [47] "node-cron," Último acceso: 2022-08-23. [Online]. Available: <https://www.npmjs.com/package/node-cron>
- [48] "Empezando | axios docs," Último acceso: 2022-08-23. [Online]. Available: <https://axios-http.com/es/docs/intro>
- [49] "Pm2 - quick start," Último acceso: 2022-08-23. [Online]. Available: <https://pm2.keymetrics.io/docs/usage/quick-start/>
- [50] "Github - expressjs/morgan: Http request logger middleware for node.js," Último acceso: 2022-08-23. [Online]. Available: <https://github.com/expressjs/morgan>
- [51] "Github - expressjs/morgan: Http request logger middleware for node.js," Último acceso: 2022-08-24. [Online]. Available: <https://ionicframework.com/docs/vue/your-first-app/deploying-mobile>
- [52] "Splash screens and icons | capacitor documentation," Último acceso: 2022-08-24. [Online]. Available: <https://capacitorjs.com/docs/guides/splash-screens-and-icons>
- [53] "Download android studio & app tools - android developers," Último acceso: 2022-08-24. [Online]. Available: <https://developer.android.com/studio>
- [54] "Carpeta compartida. visualización de datos demográficos en españa," Último acceso: 2022-09-04. [Online]. Available: https://drive.google.com/drive/folders/1tOiP3zWv1PaifWcjspWgyCYcCffSBkm_?usp=sharing
- [55] "Xcode," Último acceso: 2022-08-25. [Online]. Available: <https://apps.apple.com/es/app/xcode/id497799835?mt=12>
- [56] "Vdde," Último acceso: 2022-08-27. [Online]. Available: <https://vdde.me/menu-principal>
- [57] "Ganar dinero con la monetización de tu sitio - google adsense," Último acceso: 2022-09-1. [Online]. Available: <https://www.google.es/adsense/start/l>
- [58] "Estudio del retorno de inversión," Último acceso: 2022-08-27. [Online]. Available: <https://docs.google.com/spreadsheets/d/1qbtA0siYJnBTXglGsUc6skvqrj1RS9qfx2ABOdMsOOc/edit?usp=sharing>