



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

## Trabajo de Fin de Grado

---

**Procesamiento y visualización de datos  
abiertos en el sector turístico de Canarias**

*Processing and visualization of open data in the tourism  
sector of the Canary Islands*

Alberto Mendoza Rodríguez

---

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:

*"Procesamiento y visualización de datos abiertos en el sector turístico de Canarias"*

ha sido realizada bajo su dirección por D. **Alberto Mendoza Rodríguez**, con N.I.F. 43.487.653-N.

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

En primer lugar, me gustaría agradecer a mi familia y amigos, por apoyarme durante todos estos años. También a los compañeros que he conocido durante la etapa universitaria, con los que he vivido grandes momentos. Por último, agradecer a mi tutor Francisco, por ayudarme y guiarme en la realización de este proyecto.

# Licencia



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

## **Resumen**

*En la actualidad estamos viviendo una etapa de recuperación turística, tras la pandemia mundial iniciada en 2020, el sector se vio gravemente afectado, lo que tuvo múltiples efectos negativos, especialmente en las Islas Canarias. Este problema puso de manifiesto la necesidad de nuevas herramientas, que permitan fomentar la recuperación y digitalización del turismo. Para lograr esta recuperación, es fundamental tener una mejor comprensión del sector y de las características de los turistas, lo que es posible gracias a la gran cantidad de datos que han sido recabados durante años por las instituciones públicas. El proyecto que se presenta, pretende fomentar la mejora del sector, aportando un mayor conocimiento del ecosistema turístico canario, mediante el procesamiento y visualización de datos abiertos.*

*Este proyecto Full Stack ha consistido en desarrollar una aplicación web y una aplicación Android, que permitan visualizar, mediante gráficas, diagramas o piezas interactivas, datos de interés acerca del turismo en Canarias. Los datos originales se han obtenido de un repositorio Open Data, sin embargo, antes de ser utilizados, es necesario aplicarles un amplio procesamiento, que permita obtener datos con los que realmente se pueda generar información de valor. Una vez realizado, es posible emplear herramientas que visualicen los datos procesados, consiguiendo facilitar su comprensión y mejorar la toma de decisiones. Cabe destacar que toda la información mostrada en las aplicaciones, siempre está actualizada con los últimos datos disponibles en el repositorio.*

*Por último, se ha demostrado la viabilidad económica de un proyecto de estas características. La duración estimada de su desarrollo es de unos 9 meses, y el retorno de la inversión se obtendría aproximadamente a los 20 meses y medio de comercialización.*

**Palabras clave:** Big Data, Open Data, Visualización, Turismo, Canarias, Web, Android

## **Abstract**

*We are currently experiencing a stage of tourism recovery, after the global pandemic that began in 2020, the sector was seriously affected, which had multiple negative effects, especially in the Canary Islands. This problem highlighted the need for new tools, to promote the recovery and digitization of tourism. To achieve this recovery, it is essential to have a better understanding of the sector and the characteristics of tourists, which is possible thanks to the large amount of data that has been collected over the years by public institutions. The project presented aims to promote the improvement of the sector, providing greater knowledge of the Canarian tourism ecosystem, through the processing and visualization of open data.*

*The project has consisted of developing a web application and an Android application, which allow visualizing, through graphs, diagrams or interactive pieces, data of interest about tourism in the Canary Islands. The original data has been obtained from an Open Data repository, however, before being used, it is necessary to apply extensive processing, which allows obtaining data with is possible to generate valuable information. Once this processing has been carried out, it is possible to use tools that visualize the processed data, facilitating its compression and improving decision-making. It should be noted that all the information displayed in the applications is always updated with the latest data available in the repository.*

*Finally, the economic viability of a project of these characteristics has been demonstrated. The estimated duration of its development is about 9 months, and the return on investment would be obtained approximately 20 and a half months after marketing.*

**Keywords:** Big Data, Open Data, Visualization, Tourism, Canary Islands, Web, Android

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema . . . . .	1
1.2. Justificación . . . . .	2
1.3. Estado actual del mercado . . . . .	3
1.4. Tendencia del mercado . . . . .	5
1.5. Objetivos . . . . .	6
<b>2. Elección de arquitecturas</b>	<b>8</b>
2.1. Repositorios Open Data . . . . .	8
2.2. Visualización de datos . . . . .	10
2.3. Aplicación web . . . . .	11
2.4. Aplicación móvil . . . . .	14
2.5. Plataformas para realizar el despliegue . . . . .	16
2.6. Documentación . . . . .	17
<b>3. Desarrollo de arquitecturas</b>	<b>18</b>
3.1. Datos seleccionados . . . . .	19
3.2. Backend . . . . .	20
3.2.1. Capa de procesamiento e inserción . . . . .	20
3.2.2. Capa de obtención . . . . .	24
3.2.3. Almacenamiento de datos . . . . .	27
3.3. Frontend . . . . .	28
3.3.1. Frontend Web . . . . .	28
3.3.2. Frontend Móvil . . . . .	32
3.4. Despliegue . . . . .	34
<b>4. Estudio de viabilidad económica</b>	<b>36</b>
4.1. Funcionalidades adicionales . . . . .	36
4.2. Desarrollo del proyecto . . . . .	37
4.3. Modelo de Comercialización . . . . .	40
<b>5. Conclusiones y líneas futuras</b>	<b>44</b>
<b>6. Summary and Conclusions</b>	<b>45</b>
<b>7. Presupuesto</b>	<b>46</b>

# Índice de Figuras

1.1. Aportación del sector turístico al PIB canario entre 2010 y 2020[1] . . . . .	1
1.2. Aumento de información digital para 2025[2] . . . . .	3
1.3. Empresas que utilizan Big Data según actividad y tamaño (% total empresas)[3] . . . . .	4
1.4. Previsión ingresos Big Data en el mundo entre 2017 y 2027[4] . . . . .	6
2.1. Logotipo repositorio Canarias Datos Abiertos . . . . .	9
2.2. Visualizadores de gráficos . . . . .	10
2.3. Interés a lo largo del tiempo de los principales frameworks frontend[5] . . . . .	12
2.4. Tendencia de los principales frameworks frontend en Stack Overflow[6] . . . . .	12
2.5. Tecnologías del backend . . . . .	13
2.6. Tecnologías aplicación móvil . . . . .	16
2.7. Plataformas para realizar el despliegue . . . . .	17
3.1. Diagrama de la arquitectura general del proyecto . . . . .	18
3.2. Diagrama de la capa de procesamiento e inserción . . . . .	20
3.3. Sintaxis para programar tareas en node-cron . . . . .	23
3.4. Diagrama API REST . . . . .	25
3.5. Página de inicio aplicación web . . . . .	28
3.6. Secciones aplicación web . . . . .	29
3.7. Pie de página web . . . . .	30
3.8. Sección Número de turistas . . . . .	30
3.9. Gráfica evolución número de turistas . . . . .	31
3.10 Gráfica nacionalidades que visitan Canarias . . . . .	31
3.11 Gráfica distribución de los turistas . . . . .	32
3.12 Pantalla de inicio y menú desplegable en móvil . . . . .	33
3.13 Sección Número de turistas en móvil . . . . .	33
3.14 Visualización de gráficas en móvil . . . . .	34
3.15 Diagrama sistema desplegado . . . . .	35
4.1. Diagrama de Gantt (1) . . . . .	38
4.2. Diagrama de Gantt (2) . . . . .	39
4.3. Recursos humanos del proyecto y sus salarios . . . . .	40
4.4. Gráfico que refleja punto ROI . . . . .	43



# Índice de Tablas

2.1. Comparación librerías visualización de datos (elaboración propia) . . . . .	10
2.2. Comparación principales frameworks frontend web[7] . . . . .	11
2.3. Comparación entre aplicaciones nativas e híbridas[8] . . . . .	14
2.4. Comparación frameworks frontend móvil[9] . . . . .	15
3.1. Relación colecciones base de datos y datasets del repositorio Open Data . .	27
4.1. Comparación planes de suscripción . . . . .	41
7.1. Presupuesto del Trabajo de Fin de Grado . . . . .	46

# Capítulo 1

## Introducción

### 1.1. Definición del problema

Durante los dos últimos años, hemos vivido una pandemia mundial que ha afectado gravemente a numerosos sectores económicos, teniendo un especial impacto en el sector turístico. Esto ha provocado que uno de los principales motores de nuestro país sufra un gran retroceso, lo que se ha reflejado especialmente en las Islas Canarias, debido a la importancia del sector en el archipiélago.

En la siguiente gráfica se puede observar la gran aportación del turismo al PIB<sup>1</sup> canario:

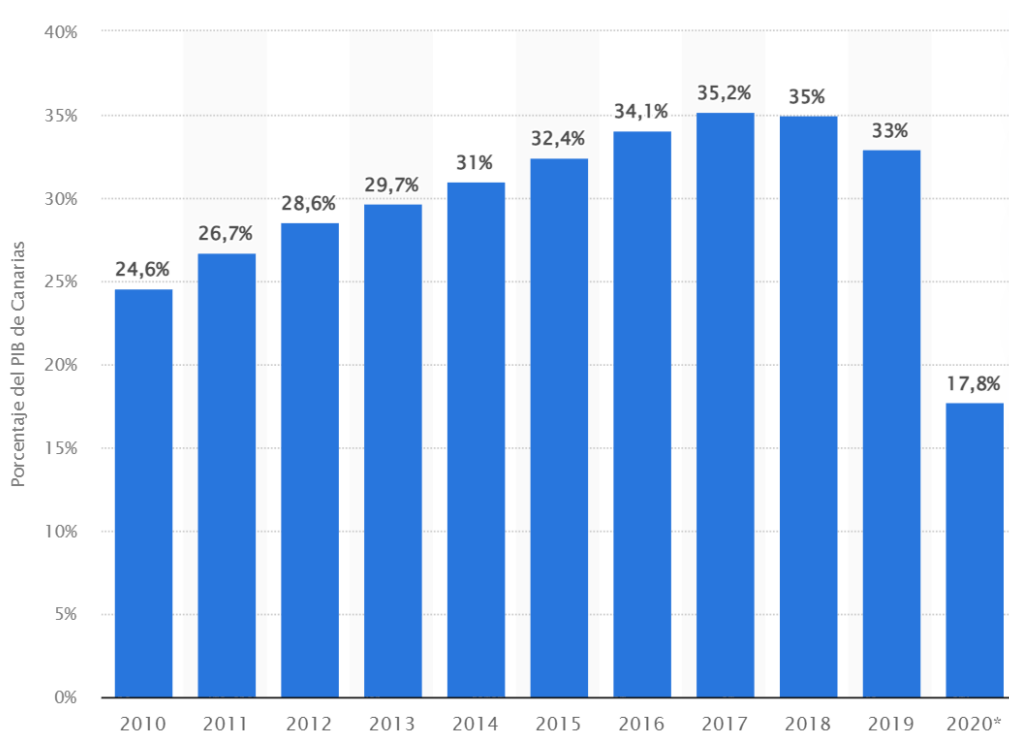


Figura 1.1: Aportación del sector turístico al PIB canario entre 2010 y 2020[1]

Como se puede ver, el peso del turismo en la economía canaria pasó del 33% en 2019 al 17,8% en 2020, tras perder más de la mitad de su cifra de negocio: 8000 millones de euros[10].

<sup>1</sup>**PIB:** el Producto Interior Bruto, o por sus siglas PIB, es un indicador económico que refleja el valor monetario de todos los bienes y servicios finales producidos por un territorio durante un tiempo específico. Se utiliza como indicador para medir la riqueza de un país.

La crisis sanitaria ha puesto de manifiesto la necesidad de acelerar la transformación digital del turismo, con el objetivo de fomentar su recuperación y continuar siendo el principal país en este ámbito. Uno de los aspectos más importantes de esta transformación digital consiste en el uso del Big Data<sup>2</sup>.

En Canarias desde principios de los años 90, se ha recopilado una ingente cantidad de datos sobre numerosos aspectos de la actividad turística, que se almacenan en bases de datos de diversas instituciones públicas (ISTAC<sup>3</sup>, SITCAN<sup>4</sup>, Ayuntamientos, . . . ), sin ser utilizados para generar mayor conocimiento.

Por ello, ante la situación descrita, surge la necesidad de crear nuevas herramientas de análisis turístico, que permitan una gestión eficiente del inmenso conjunto de datos disponible, logrando con esto fomentar y renovar el sector turístico.

## 1.2. Justificación

El prototipo de aplicación que se presenta en este proyecto puede ayudar a fomentar y renovar el turismo canario por los siguientes factores:

- **Open Data**<sup>5</sup>: todos los datos que se utilizan proceden de instituciones públicas, lo que permite disponer de una gran masa de datos sobre el turismo. Además, hace posible que los consumidores finales de la aplicación tengan la seguridad de que la información, generada tras el procesamiento de los datos, procede de una fuente fiable.
- **Datos en tiempo real**: la aplicación muestra en todo momento información generada con los últimos datos disponibles en el repositorio Open Data, esto hace que los usuarios conozcan el resultado de sus estrategias más rápidamente, agiliza la toma de decisiones, facilita la observación de cambios y refleja la tendencia actualizada del sector.
- **Visualización de datos**: el uso de elementos visuales como gráficos, diagramas o piezas interactivas, ofrece de una manera más accesible la posibilidad de detectar y comprender las tendencias, valores atípicos y patrones en los datos. Esta alternativa visual incorporada en la aplicación, facilita la asimilación de la información generada tras el procesamiento de los datos en bruto.

Los tres factores descritos permiten generar un prototipo de aplicación que proporciona una mejor visión del sector turístico en Canarias, centrando esta visión en cuatro factores fundamentales: número de turistas, gasto, estancia y expectativas. Con esta información de valor, visualizada de forma atractiva y actualizada con los últimos datos disponibles, será posible un mayor conocimiento del ecosistema turístico canario.

---

<sup>2</sup>**Big Data**: término que hace referencia al análisis masivo de datos, que son tan sumamente grandes, rápidos y complejos, que las aplicaciones de software de procesamiento tradicionales no son capaces de gestionarlos en un tiempo razonable.

<sup>3</sup>**ISTAC**: Instituto Canario de Estadística, es el órgano central del sistema estadístico autonómico y centro oficial de investigación del Gobierno de Canarias.

<sup>4</sup>**SITCAN**: Sistema de Información Territorial de Canarias, se trata del sistema de información corporativo de la Comunidad Autónoma de Canarias que produce, integra, gestiona y difunde conocimiento e información del territorio.

<sup>5</sup>**Open Data**: filosofía y práctica que persigue que los datos que recogen, producen y difunden las administraciones públicas estén disponibles libremente.

### 1.3. Estado actual del mercado

En los últimos años la información disponible a nivel mundial ha aumentado de manera exponencial, lo que ha provocado que el Big Data se convierta en un concepto de creciente popularidad. Según el World Economic Forum se estima que 44 zettabytes<sup>6</sup> conformaron en 2020 todo el universo digital, y se espera que esta cantidad de datos siga en aumento, ya que en 2025 se estima que se crearán 463 exabytes<sup>7</sup> (0,463 zettabytes) de datos cada día en todo el mundo[11]. Sin embargo, esta gran cantidad de datos pierde valor sin un tratamiento adecuado, por ello la verdadera importancia del Big Data reside en la capacidad de poder extraer información y conocimiento de valor.

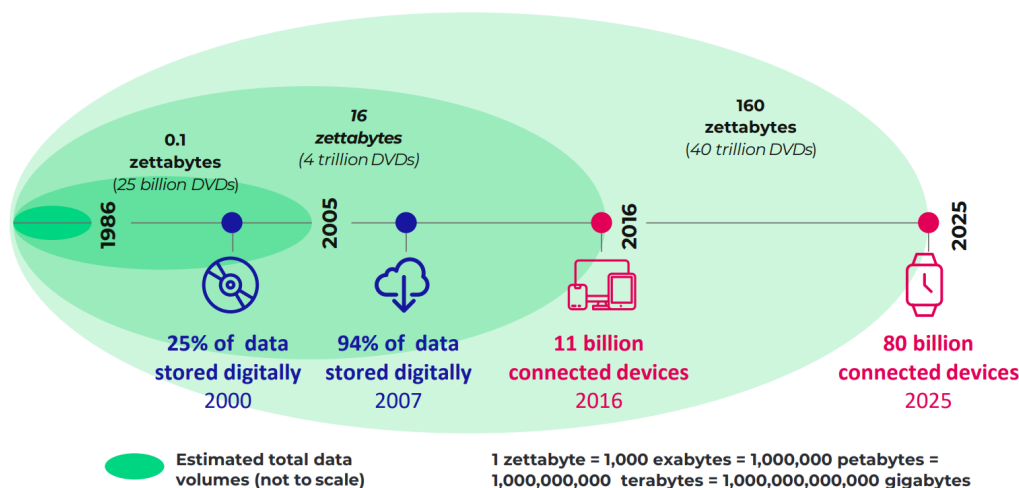


Figura 1.2: Aumento de información digital para 2025[2]

Con el objetivo de tener una mayor compresión del Big Data, se suele emplear el Modelo de las 3 V, ideado en 2001 por el analista de Gartner<sup>8</sup>, Doug Laney. De esta forma las 3 V del Big Data son[12, 13]:

- **Volumen:** ha de ser capaz de gestionar un gran volumen de datos, que se generan y recopilan constantemente. Por ello, se trata de uno de los principales retos, al requerir capacidades de almacenamiento y procesamiento que superan las características de los equipos y sistemas tradicionales.
- **Velocidad:** es necesario que pueda almacenar y trabajar en tiempo real con las fuentes generadoras de información. Esto aporta un gran valor a las empresas al proporcionar los datos en el momento adecuado para la toma de decisiones.
- **Variedad:** debe tener la capacidad de combinar una gran variedad de información en los diferentes formatos en las que se puede presentar. El auge de las tecnologías ha provocado la aparición de nuevos tipos de datos no estructurados, que requieren un preprocesamiento adicional para aportar significado y habilitar los metadatos<sup>9</sup>.

<sup>6</sup>**Zettabytes:** unidad de almacenamiento de información que equivale a  $10^{21}$  bytes.

<sup>7</sup>**Exabytes:** unidad de almacenamiento de información que equivale a  $10^{18}$  bytes.

<sup>8</sup>**Gartner:** se trata de una de las empresas consultoras y de investigación de las tecnologías de la información más importantes del mundo. Se dedica al análisis de las tendencias del mercado.

<sup>9</sup>**Metadatos:** son datos que describen otros datos, por tanto, aportan información sobre los datos contenidos en algún objeto denominado recurso.

Por tanto, a pesar de los grandes beneficios que aporta el Big Data, su adopción en las organizaciones y el desarrollo de proyectos en este ámbito presenta importantes desafíos.

En los últimos años, la relevancia del Big Data ha experimentado un constante crecimiento, lo que se refleja en el estudio *Worldwide Big Data and Analytics Spending Guide*[14] realizado por la corporación IDC<sup>10</sup>, donde se estima que en 2017 los ingresos totales del sector en España fueron más de 170 millones de euros, en 2018 llegaron a los 218 millones y en 2019 alcanzaron los 251 millones de euros.

Uno de los factores que ha impulsado la adopción del Big Data es la reducción de los costes de almacenamiento y de procesamiento de datos, tal y como afirma el estudio acerca del *Uso de tecnologías digitales por empresas en España 2022*[3], realizado por el Observatorio Nacional de Tecnología y Sociedad (ONTSI). En este estudio también se muestra que el 11 % de las empresas en general en nuestro país ya aplican técnicas de Big Data, en concreto estas técnicas son utilizadas por el 29 % de las grandes empresas, el 18 % de las medianas y el 9 % de las pequeñas.

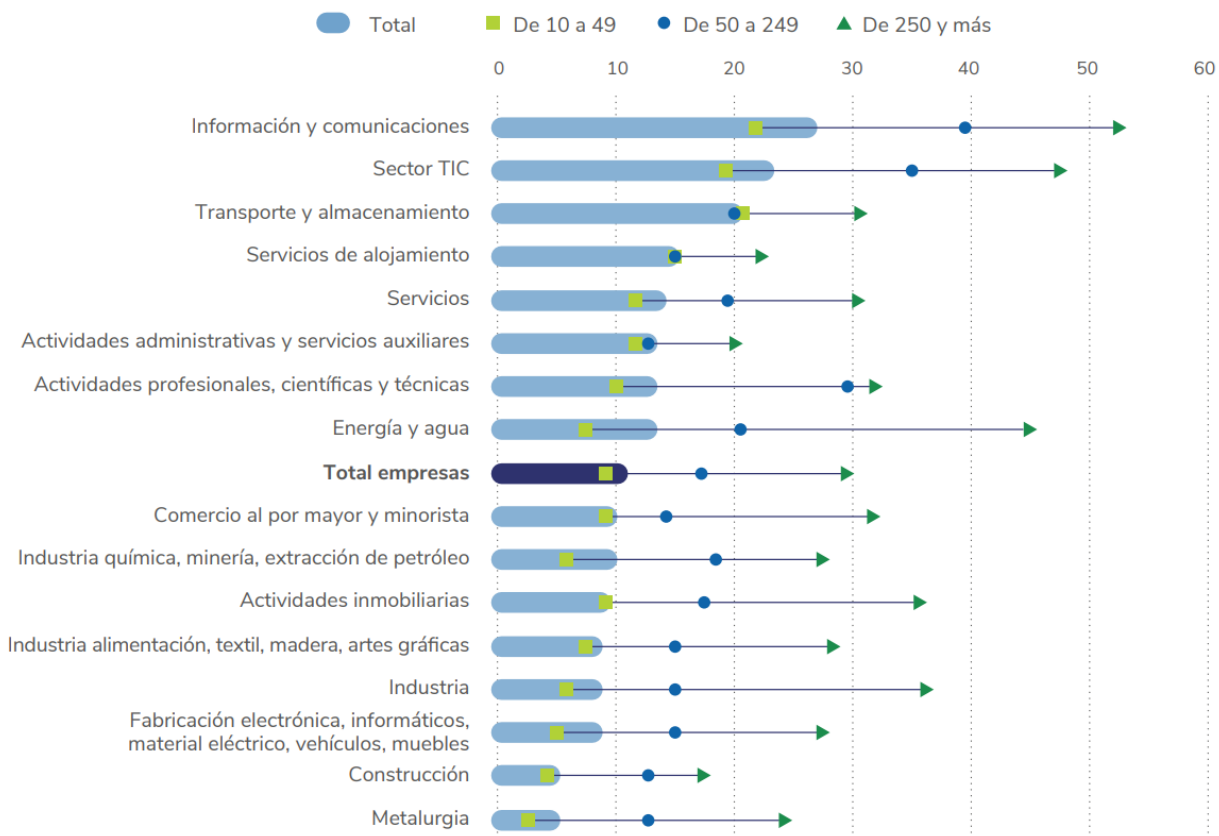


Figura 1.3: Empresas que utilizan Big Data según actividad y tamaño (% total empresas)[3]

<sup>10</sup>IDC: : siglas de International Data Corporation, es el principal proveedor mundial de inteligencia de mercado, servicios de asesoramiento y eventos para los mercados de tecnología de la información, telecomunicaciones y tecnología de consumo.

Una vez observado que el Big Data se trata de un concepto de gran relevancia en la actualidad y con una proyección creciente, resulta de interés estudiar algunas soluciones disponibles en el mercado que aplican técnicas de Big Data al sector turístico. A continuación, se muestran los proyectos más destacados:

- **Quality Focus Online**[15]: herramienta web de la multinacional NH Hotel Group, que gestiona toda la información publicada en Internet sobre la satisfacción y opinión del cliente acerca de la cadena hotelera. Esto lo ha convertido en una herramienta fundamental, utilizada diariamente por los gestores de hoteles para la toma de decisiones.
- **Smart Steps**[16]: plataforma Big Data de Telefónica que permite a las empresas obtener información acerca de los turistas que llegan a la ciudad, su origen y su desplazamiento. Para lograr esto, Telefónica utiliza su gran red móvil donde se producen diariamente miles de millones de eventos, de los cuales puede extraer información sobre las tendencias globales de grupos de personas.
- **Mabrian**[17]: plataforma Big Data centrada exclusivamente en el sector turístico, creada por la compañía Mabrian Technologies S.L., analiza en tiempo real múltiples fuentes de datos de manera simultánea para realizar un análisis multivariable. De esta forma es capaz de analizar el ciclo completo del viajero desde el periodo de búsqueda inicial, hasta la percepción final de la experiencia, aportando una visión completa del ciclo del turista.
- **ForwardKeys**[18]: es una de las principales compañías de análisis de datos turísticos, permite a sus clientes conocer quién viaja, dónde, cuándo y por cuánto tiempo. Esto es posible gracias a que poseen unas de las bases de datos de viajes aéreos más completas del sector, incluye información sobre los billetes reservados tanto a través de las agencias de viajes, como directamente de las aerolíneas. Con esta gran cantidad de datos son capaces de informar en tiempo real sobre los patrones de viaje, la demanda, las preferencias de los distintos públicos, etc. Disponen de tres plataformas online: *Destination Gateway*, *Traveller Statistics* y *ForwardKeys Nexus*.
- **Dataestur**[19]: portal de datos del turismo en España que permite consultar varias fuentes de datos de interés. Nace de la iniciativa de SEGITTUR<sup>11</sup>, con el objetivo de ofrecer un punto único de referencia donde consultar información sobre el turismo de nuestro país, además, actualiza su contenido a medida que lo hace la fuente oficial. Incluye 5 categorías principales: datos generales, economía, transporte, alojamiento y conocimiento.

## 1.4. Tendencia del mercado

Como ya se ha comentado, el Big Data es un concepto de gran importancia en la actualidad y con una enorme proyección, en los próximos años se espera que aumente la cantidad de empresas que apliquen técnicas de analítica de datos, así como el número de aplicaciones y proyectos en este sector. Todo ello hace que la previsión de ingresos provenientes del Big Data tenga una tendencia creciente.

<sup>11</sup>**SEGITTUR**: Sociedad Mercantil Estatal para la Gestión de la Innovación y las Tecnologías Turísticas (S.A.M.), dependiente del Ministerio de Industria, Comercio y Turismo, y adscrita a la Secretaría de Estado de Turismo.

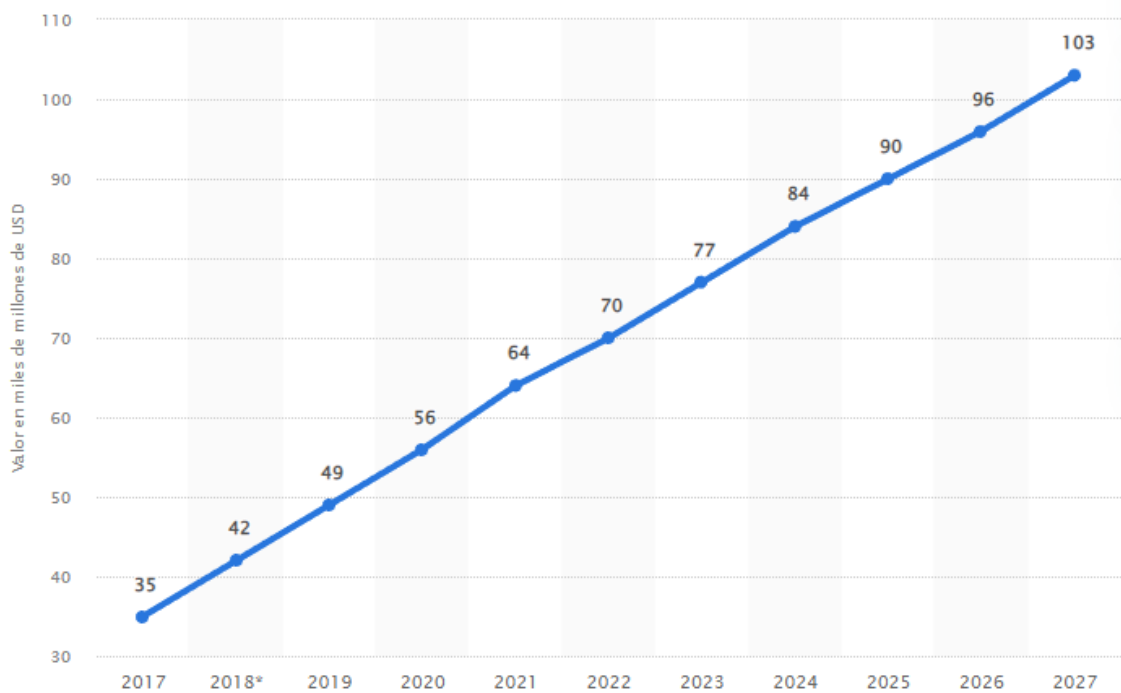


Figura 1.4: Previsión ingresos Big Data en el mundo entre 2017 y 2027[4]

Además, también se prevé un aumento en la creación de proyectos innovadores, que tengan como objetivo intensificar la digitalización del sector turístico, impulsado especialmente por el plan de modernización y competitividad del sector turístico financiado con los fondos Next Generation EU[20].

## 1.5. Objetivos

El estudio que se ha realizado pretende responder a la siguiente pregunta: **¿Es viable una aplicación web y una aplicación móvil que permitan la visualización de datos en tiempo real sobre el sector turístico?**

Para responder a la pregunta planteada se han definido los siguientes objetivos:

- Analizar las soluciones actuales que ofrece el mercado en cuanto al uso de técnicas Big Data en el sector turístico.
- Estudiar las herramientas y tecnologías disponibles para desarrollar el proyecto, seleccionando aquellas que mejor se adapten a los requisitos del mismo.
- Desarrollar arquitecturas:
  - Desarrollar Backend.
    - Implementar sistema para el procesamiento periódico de los datos disponibles en el repositorio Open Data.
    - Implementar API REST que facilite el acceso a los datos procesados.
  - Desarrollar Frontend Web y Móvil que permitan una adecuada visualización de los datos generados tras el procesamiento.

- Interconectar arquitecturas.
- Comprobar el correcto funcionamiento de la aplicación web y de la aplicación móvil.
- Desarrollar un plan de negocio para la comercialización del producto:
  - Crear Diagrama de Gantt<sup>12</sup> en el que se definan las tareas a realizar, sus duraciones, sus recursos y sus costes.
  - Prever el coste total del proyecto.
  - Diseñar un modelo de comercialización del producto.
  - Calcular el ROI<sup>13</sup>.

---

<sup>12</sup>**Diagrama de Gantt:** herramienta gráfica para planificar proyectos, que proporciona una vista general de las tareas programadas y el tiempo de dedicación previsto para cada una de ellas. No indica las relaciones entre las tareas.

<sup>13</sup>**ROI:** el retorno de la inversión (ROI por las siglas en inglés *return on investment*) es una métrica que mide el rendimiento económico de una inversión, para ello calcula a partir de qué punto las ganancias superan la inversión inicial realizada.



# Capítulo 2

## Elección de arquitecturas

En esta sección se explica el estudio previo que se ha realizado sobre las posibles herramientas y tecnologías, que se pueden utilizar para desarrollar el proyecto. Además, se detallan aquellas que finalmente se han decidido emplear.

Cabe destacar que de forma general para desarrollar el proyecto se ha utilizado el editor de código fuente Visual Studio Code, y como sistema de control de versiones se ha empleado Git<sup>1</sup> y GitHub<sup>2</sup>.

### 2.1. Repositorios Open Data

En primer lugar, al tratarse de una aplicación cuya funcionalidad se basa en el procesamiento y visualización de datos, es fundamental que la fuente de estos datos sea de calidad. Para lograrlo se ha realizado un análisis de los repositorios Open Data disponibles, aplicando los siguientes criterios de calidad:

- **Acceso fácil y gratuito:** tiene que existir un mecanismo que facilite el consumo de los datos que conforman el repositorio, además, este uso debe ser gratuito.
- **Guía de uso sencilla:** el repositorio debe ofrecer junto con los conjuntos de datos información sobre los mismos.
- **Fiabilidad:** es necesario que el repositorio posea las acreditaciones y reconocimientos necesarios que validen la adecuada gestión de los datos.
- **Gestión a largo plazo:** es fundamental garantizar que el funcionamiento y mantenimiento del repositorio van a perdurar en el tiempo.
- **Formatos comunes:** los conjuntos de datos deben poder ser descargados o exportados en formatos ampliamente utilizados.
- **Precisión de los datos:** la aplicación está destinada a visualizar información en tiempo real sobre el turismo en Canarias, por ello es imprescindible que el repositorio disponga de suficientes datos en este ámbito, y que estos se actualicen frecuentemente.

---

<sup>1</sup>**Git:** es un sistema de control de versiones distribuido gratuito y de código abierto. Su propósito es llevar registro de los cambios, incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

<sup>2</sup>**GitHub:** es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git, permitiendo trabajar en proyectos de software desde cualquier lugar.

Tras analizar diferentes repositorios Open Data bajo los criterios anteriormente descritos, destacan dos repositorios que se pueden utilizar para desarrollar el proyecto:

- **datos.gob.es**[21]: plataforma asociada a la Iniciativa Aporta[22], que se trata de una estrategia nacional, iniciada en 2009, para fomentar la apertura de datos por parte de las administraciones públicas. Por tanto, esta plataforma proporciona un punto de acceso a los conjuntos de datos de los diferentes niveles administrativos (nacional, autonómico, local, etc). Además, ofrece información general, materiales formativos y noticias de actualidad, todo ello relacionado con la reutilización de información del sector público.
- **Canarias Datos Abiertos**[23]: plataforma creada en 2021, gracias al impulso realizado por la Dirección General de Transparencia y Participación Ciudadana del Gobierno de Canarias, en colaboración con la Dirección General de Telecomunicaciones y Nuevas Tecnologías. Este portal sustituye al anterior *opendata.gobiernodecanarias.org*, para añadir nuevas funcionalidades e incrementar significativamente la cantidad de conjuntos de datos disponibles. Con ello se persigue el objetivo de convertir a esta nueva plataforma, en el único punto de acceso a los datos abiertos de Canarias.

Ambos repositorios cumplen con los criterios de calidad que se establecieron al inicio del análisis. Finalmente, se ha decidido emplear el repositorio **Canarias Datos Abiertos**, al considerar que existirá una mayor cantidad y una actualización más frecuente de aquellos conjuntos de datos relacionados con el turismo canario.



Figura 2.1: Logotipo repositorio Canarias Datos Abiertos

Este repositorio incluye más de 7600 conjuntos de datos, haciendo que sea el punto de acceso único con más datos de España. Esto es posible gracias a la unión de los datos procedentes del Gobierno de Canarias, Instituto de Estadística de Canarias (ISTAC) y del Sistema de Información Territorial de Canarias (SITCAN). A continuación, se detallan las características técnicas más relevantes de este repositorio:

- Los datos se clasifican en 4 categorías: datos por tema (22 grupos), datos por organización (15 organizaciones), datos más recientes y datos más populares.
- Utiliza la herramienta de código abierto CKAN<sup>3</sup>, bajo licencia GNU<sup>4</sup>, lo que facilita el consumo y reutilización de los conjuntos de datos.
- Los formatos más comunes para la descarga y exportación de los datos son: JSON, CSV, GeoJSON, GPKG y HTML.

<sup>3</sup>**CKAN**: Comprehensive Knowledge Archive Network, es una aplicación web de código abierto para el almacenamiento y la distribución de los datos.

<sup>4</sup>**GNU**: General Public License, es una licencia de derecho de autor que garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

## 2.2. Visualización de datos

Una de las etapas más relevantes del procesamiento de datos consiste en la interpretación de los mismos, lo cual sólo es posible si estos datos son utilizables por los usuarios. Entre las soluciones disponibles para presentar adecuadamente los resultados del procesamiento, destaca la visualización de los datos, que facilita comprender la información y detectar patrones y tendencias.

En el mercado existen numerosas librerías para diferentes lenguajes de programación, que proporcionan funcionalidades para visualizar datos. Sin embargo, el análisis se centrará en aquellas construidas mediante JavaScript<sup>5</sup>, ya que será el lenguaje utilizado para implementar el frontend.

A continuación, se muestra una tabla comparativa de las tres principales librerías de visualización de datos que se han considerado: **Highcharts**[24], **D3.js**[25] y **Chart.js**[26].

	<b>Highcharts</b>	<b>D3.js</b>	<b>Chart.js</b>
Licencia de uso	Gratuito para usos no comerciales <sup>6</sup>	Open Source	Open Source
Formato de gráficos	Canvas <sup>7</sup> / SVG <sup>8</sup> / VML <sup>9</sup>	SVG / Canvas	Canvas
Curva de aprendizaje	Moderada	Elevada	Baja
Variedad de gráficos	Numerosos gráficos	Numerosos gráficos	Limitada (8 tipos de gráfico)
Nivel de personalización	Alto	Alto	Bajo

Tabla 2.1: Comparación librerías visualización de datos (elaboración propia)

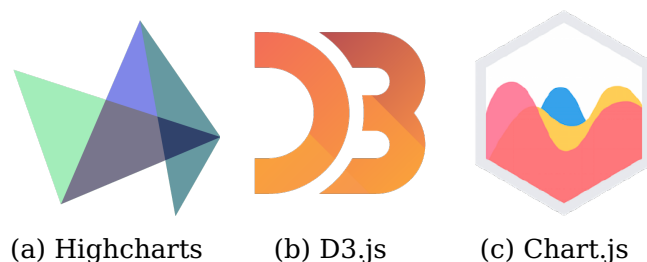


Figura 2.2: Visualizadores de gráficos

<sup>5</sup>**JavaScript**: es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente para desarrollar aplicaciones web.

<sup>6</sup>Proyectos personales, sitios web educativos y organizaciones sin fines de lucro.

<sup>7</sup>**Canvas**: elemento HTML incorporado en HTML5 que permite la generación de gráficos dinámicamente.

<sup>8</sup>**SVG**: *scalable vector graphics*, es un lenguaje de marcado XML para describir gráficos vectoriales en dos dimensiones. En estos gráficos la calidad de imagen no está afectada por la resolución de las pantallas.

<sup>9</sup>**VML**: lenguaje XML de programación abierto, destinado a la creación de gráficos vectoriales.

Por tanto, con la tabla comparativa se puede concluir que Highcharts y D3.js, son las dos librerías que presentan mejores características para ser utilizadas en el proyecto. Se descarta Chart.js debido a su limitada variedad de gráficas. Finalmente, se determinó que la mejor solución consiste en utilizar Highcharts, ya que proporciona una mayor velocidad de desarrollo en comparación a D3.js, gracias a su menor complejidad.

En cuanto a las características que presenta Highcharts, destaca la facilidad de implementar una gran variedad de gráficos con alta interactividad, la capacidad de renderizarlos en formato SVG, y la posibilidad de incluir módulos para potenciar sus funcionalidades.

## 2.3. Aplicación web

En cuanto al desarrollo de la aplicación web esta se divide en dos partes claramente diferenciadas:

- **Frontend (lado del cliente):** se trata de la parte de la aplicación que interactúa con los usuarios, es decir, aquello que se visualiza en pantalla.
- **Backend (lado del servidor):** es la parte oculta de la aplicación, encargada de gestionar la lógica de la misma.

Por tanto, se tiene que realizar un análisis de las tecnologías a utilizar en el lado del cliente y del servidor, con el objetivo de generar un stack de tecnologías<sup>10</sup> que proporcione una solución completa para implementar el proyecto.

Comenzando en el lado del cliente los principales frameworks frontend son: **React**[27], **Angular**[28] y **Vue**[29]. A continuación, se muestra una tabla con las características más destacadas de estos frameworks:

	<b>React</b>	<b>Angular</b>	<b>Vue.js</b>
Desarrollado por	Facebook	Google	Evan You
Tipo	Librería JavaScript	Framework web	Framework JavaScript
Año de lanzamiento	2013	2010	2014
Lenguaje de programación	JavaScript	TypeScript <sup>11</sup>	JavaScript
DOM <sup>12</sup>	Virtual DOM <sup>13</sup>	Real DOM	Virtual DOM
Curva de aprendizaje	Moderada	Elevada	Moderada
Velocidad de desarrollo	Normal	Lento	Rápido

Tabla 2.2: Comparación principales frameworks frontend web[7]

<sup>10</sup>**Stack de tecnologías:** se trata de una combinación de tecnologías que se complementan entre sí, para crear de manera efectiva un proyecto de software.

<sup>11</sup>**TypeScript:** es un lenguaje de programación que se trata de un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases.

<sup>12</sup>**DOM:** es una interfaz que trata un documento HTML como una estructura de árbol, donde cada nodo es un objeto que representa una parte del documento.

<sup>13</sup>**Virtual DOM:** es una forma de representar el DOM real de una página web con objetos Javascript, consiguiendo simplificar el seguimiento de las actualizaciones en tiempo real.

En cuanto a la popularidad de los frameworks destaca significativamente React.

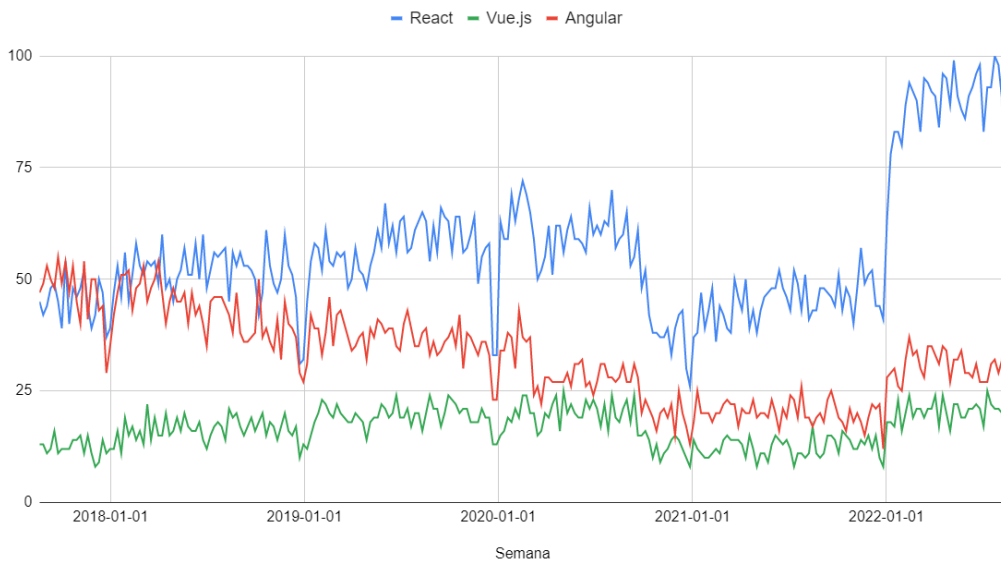


Figura 2.3: Interés a lo largo del tiempo de los principales frameworks frontend[5]

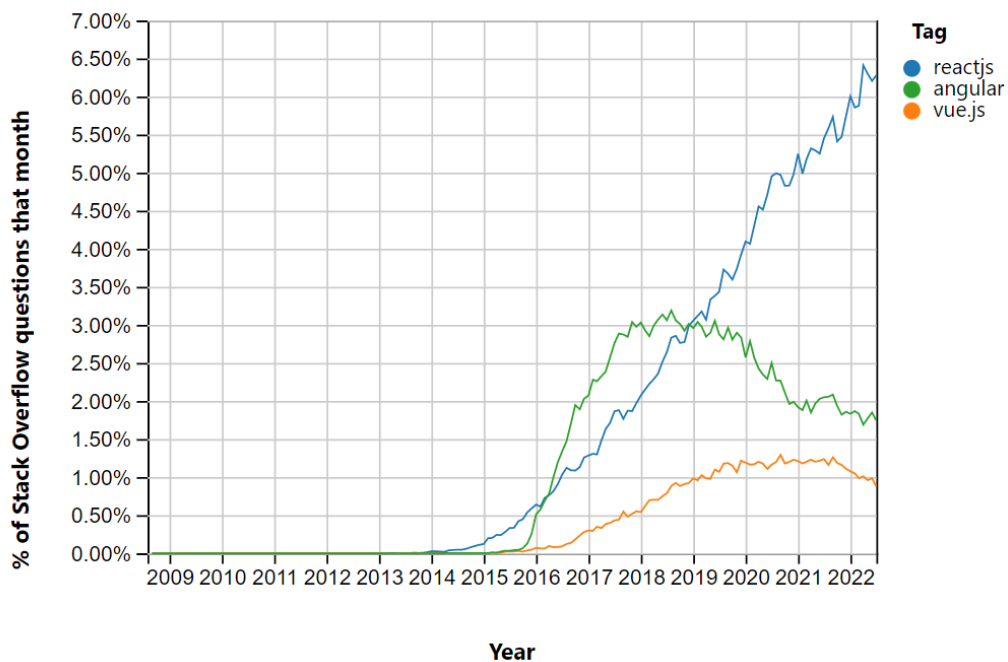


Figura 2.4: Tendencia de los principales frameworks frontend en Stack Overflow[6]

Tras el análisis se ha decidido trabajar con **React**, ya que, además de ser el más popular, es más sencillo de aprender que Angular, emplea JSX<sup>14</sup> que consigue una sintaxis más corta y simple, ofrece una optimización superior en el renderizado gracias al DOM virtual, genera aplicaciones SPA<sup>15</sup> y permite la reutilización de sus componentes.

<sup>14</sup>**JSX**: extensión de la sintaxis de JavaScript, que proporciona una forma de estructurar la representación de componentes empleando una sintaxis similar en apariencia a HTML.

<sup>15</sup>**SPA**: *Single-Page Application*, es un tipo de aplicación web que ejecuta todo su contenido en una sola página.

En cuanto al lado del servidor existe un requisito que ha de cumplir el framework elegido, consiste en que debe permitir el uso de TypeScript para implementar el backend. Resulta de especial interés utilizar este lenguaje de programación basado en JavaScript, ya que es fuertemente tipado lo que reducirá el número de errores en el procesamiento de los datasets<sup>16</sup>. Además, otro factor determinante es que disminuirá la velocidad de desarrollo, gracias a la experiencia que se tiene en el mismo.

Por ello, se ha decidido trabajar con Node.js y Express para implementar el backend:

- **Node.js**[30]: se trata de un entorno de ejecución multiplataforma y de código abierto para ejecutar JavaScript fuera de un navegador, que permite la creación de aplicaciones rápidas y escalables. Cabe destacar que utiliza un modelo de E/S sin bloqueo y basado en eventos que lo hace liviano y eficiente, perfecto para aplicaciones en tiempo real con uso intensivo de datos que se ejecutan en dispositivos distribuidos.
- **Express**[31]: se trata de un framework web minimalista y flexible que se asienta sobre la funcionalidad del servidor web de Node.js, para proporcionar un conjunto sólido de nuevas funcionalidades. Entre estas funcionalidades destaca un enrutamiento simple para las solicitudes realizadas por los clientes, y un middleware que se encarga de tomar decisiones para dar respuestas correctas. De esta forma se facilita el desarrollo de aplicaciones basadas en Node.

Por tanto, se utilizará el potente rendimiento de Node.js y la facilidad de codificación de Express para implementar un backend de calidad.

La base de datos que almacenará los datos resultantes del procesamiento debe ser de tipo no relacional. Esto se debe a que es necesario definir modelos de datos específicos, que no tengan una relación entre ellos, pues cada dataset procedente del repositorio Open Data será independiente del resto. Además, este tipo de base de datos ofrece mayor flexibilidad, al manejar fácilmente cualquier formato de datos, una alta escalabilidad y rendimiento gracias a su escalado horizontal<sup>17</sup>, y una elevada disponibilidad.

Se ha decidido emplear **MongoDB**[32], base de datos no relacional más relevante en la actualidad, que almacena datos en documentos flexibles similares a JSON, estos documentos se organizan en grupos lógicos denominados colecciones. La implementación de MongoDB se realizará mediante **MongoDB Atlas**, base de datos en la nube donde se gestiona la complejidad de todas sus implementaciones en el proveedor de servicios en la nube, en el caso del proyecto este proveedor es AWS<sup>18</sup>.



Figura 2.5: Tecnologías del backend

<sup>16</sup>**Datasets:** colección de datos habitualmente tabulada en un sistema de almacenamiento de datos.

<sup>17</sup>**Escalado horizontal:** consiste en potenciar el rendimiento del sistema desde un aspecto de mejora global, a diferencia de aumentar la potencia de una única parte del mismo. Este modelo implica tener varios servidores trabajando como un todo.

<sup>18</sup>**AWS:** Amazon Web Services, es una plataforma en la nube que ofrece servicios para la gestión de datos.

Tras establecer los frameworks que se van a utilizar en el lado del cliente y del servidor, así como la base de datos, se concluye que finalmente se empleará el **stack MERN** (MongoDB, Express, React y Node.js) para el desarrollo de la aplicación web.

También cabe destacar otras tecnologías que se han utilizado:

- **React-Bootstrap**[33]: utilizada en el frontend, se trata de una librería basada en componentes que proporciona Bootstrap nativos como componentes React puros.
- **Axios**[34]: empleada en el backend, es un cliente HTTP basado en promesas para Node.js y el navegador, permite realizar las peticiones al repositorio Open Data para consumir los datasets deseados.
- **node-cron**[35]: es una librería JavaScript que permite programar tareas para que se ejecuten en momentos o intervalos específicos, es necesaria en el backend para realizar peticiones periódicas al repositorio Open Data.
- **mongoose**[36]: requerida en el backend al ser el principal ODM<sup>19</sup> para MongoDB.

## 2.4. Aplicación móvil

En la aplicación móvil se utiliza el mismo backend que en la aplicación web, por tanto las tecnologías empleadas para el lado del servidor coinciden con las descritas en la sección anterior.

En cuanto al lado del cliente existen dos tipos de aplicaciones para llevar a cabo el desarrollo:

- **Aplicaciones nativas:** se construyen utilizando entornos de desarrollo integrados (IDE) y lenguajes para sistemas operativos móviles como Apple iOS o Google Android. Permiten un acceso completo al dispositivo, pero pueden ser más costosas y complejas que otras tecnologías.
- **Aplicaciones híbridas:** son aplicaciones web que actúan como aplicaciones nativas. Se desarrollan utilizando tecnologías como HTML, JavaScript y CSS. En comparación con las aplicaciones nativas permiten un desarrollo más rápido y rentable, sin embargo, no ofrecen tantas funcionalidades.

	<b>Aplicaciones nativas</b>	<b>Aplicaciones híbridas</b>
Coste de desarrollo	Alto	Medio
Tiempo de desarrollo	Alto	Medio
Multiplataforma	No	Sí
Rendimiento	Alto	Medio
Apps Stores	Sí	Sí
Acceso al dispositivo	Completo	Alto
Conexión a Internet	No siempre necesario	No siempre necesario
Espacio en el dispositivo	Sí	Sí

Tabla 2.3: Comparación entre aplicaciones nativas e híbridas[8]

<sup>19</sup>**ODM:** Object Document Mapping, permite traducir el formato de los objetos utilizados en el código, a la representación empleada en el documento de la base datos no relacional.

Teniendo en cuenta las necesidades del proyecto, resulta más interesante el desarrollo híbrido. Esto se debe a que la aplicación desarrollada mostrará en pantalla la información que se encuentra almacenada en la base de datos, por tanto, no es necesario disponer de complejas funcionalidades nativas. Además, son más sencillas, económicas y permiten la reutilización de código. Por tanto, se reducirán los tiempos de desarrollo y se conseguirá un diseño más homogéneo entre las aplicaciones web y móvil, al poder adaptar fácilmente fragmentos del código del frontend.

Los principales frameworks para el desarrollo de aplicaciones móviles multiplataforma, dentro del ecosistema JavaScript, son **Ionic**[37] y **React Native**[38].

	<b>Ionic</b>	<b>React Native</b>
Stack de tecnologías	Conocimientos básicos de HTML/CSS, JavaScript y cualquier framework de frontend.	Conocimiento obligatorio de JavaScript y React.js
Reutilización de código	El mismo código para crear aplicaciones para cualquier plataforma, dispositivo y sistema operativo.	El mismo código base se puede usar para dispositivos Android e iOS, sin embargo, es posible que se necesite una pequeña parte del código específico de la plataforma.
Pruebas de código	Las pruebas se pueden realizar directamente en el navegador.	Es posible que se necesite un emulador o dispositivo para probar correctamente.
Performance	Las aplicaciones son un poco más lentas.	El rendimiento es lo más cercano posible al nativo. También se utilizan algunos de los componentes nativos.
Comunidad	Muy solidario pero más débil.	Enorme, desarrollado con Facebook y su comunidad.

Tabla 2.4: Comparación frameworks frontend móvil[9]

Junto con este análisis, se llevaron a cabo pruebas para comprobar la integración de la librería de visualización de datos Highcharts con ambos frameworks. Se obtuvo un resultado claro, la integración con React Native produjo numerosos problemas, debido a que el mantenimiento del wrapper<sup>20</sup> que ofrece Highcharts para este framework, se suspendió en enero de 2021. Esto implicaba numerosos problemas de compatibilidad con las versiones actuales de React Native. En cambio, con Ionic se experimentó una gran facilidad de integración, al poder utilizar directamente el mismo wrapper empleado en React.

<sup>20</sup>**Wrapper:** término que hace referencia a programas o códigos que rodean otros componentes de programa, a menudo se utilizan para mejorar la compatibilidad entre diferentes estructuras de software.



En concreto se utiliza **Ionic React** que se trata de la versión React nativa del framework Ionic. El lenguaje empleado será TypeScript, al ser el utilizado por defecto en Ionic. Para construir finalmente la aplicación móvil, también será necesario emplear el IDE **Android Studio**<sup>21</sup>[39].

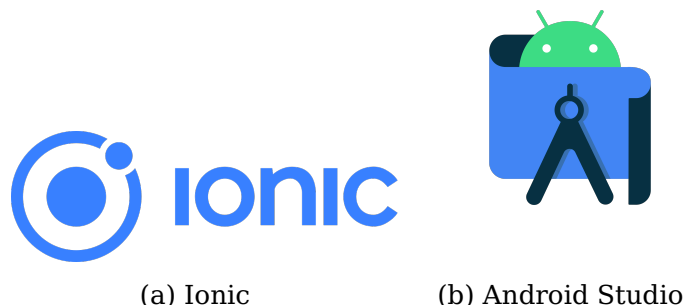


Figura 2.6: Tecnologías aplicación móvil

## 2.5. Plataformas para realizar el despliegue

Una vez finalizada la implementación de la aplicación web, será necesario desplegar el software para que el sistema esté disponible para los usuarios. Existen numerosas plataformas basadas en la nube que agilizan este proceso, al facilitar las configuraciones en infraestructura, ya que evitan la necesidad de crear servidores propios.

Entre las plataformas disponibles en el mercado para llevar a cabo el despliegue, se han considerado las siguientes:

- **Heroku**[40]: es una PaaS<sup>22</sup> basada en la nube que soporta múltiples lenguajes de programación. Utiliza contenedores Linux, llamados dynos, para alojar las aplicaciones. También cuenta con la posibilidad de añadir nuevas funcionalidades mediante los denominados add-ons, los cuales son complementos que permiten incluir servicios ofrecidos por otras compañías. Además, en el momento de desarrollar este prototipo, Heroku ofrece un plan gratuito para aplicaciones no comerciales.
- **DigitalOcean**[41]: es un proveedor de servicios de computación en la nube e IaaS<sup>23</sup>, entre los múltiples servicios que proporciona, destaca el alojamiento de aplicaciones y servicios web. Para ello, emplea máquinas virtuales basadas en Linux que la compañía denomina droplets. Los desarrolladores tienen la capacidad de elegir el tamaño del droplet, así como la región geográfica y data center donde será ejecutado. Además, ofrece a los estudiantes créditos gratuitos de \$100 para desarrollar sus proyectos.

---

<sup>21</sup>**Android Studio**: es el entorno de desarrollo integrado oficial para la plataforma Android.

<sup>22</sup>**PaaS**: *Platform as a service*, se trata de una plataforma donde un proveedor de servicios ofrece acceso a un entorno basado en la nube, en el cual los usuarios pueden crear y distribuir aplicaciones.

<sup>23</sup>**IaaS**: *Infrastructure as a service*, se trata de una plataforma donde el proveedor proporciona a los clientes acceso de pago por uso de almacenamiento, las redes, los servidores y otros recursos en la nube.

Tras analizar ambas plataformas, se concluyó que la mejor solución consistía en utilizar **DigitalOcean**. Esto se debe a que el plan gratuito ofrecido por Heroku, tiene ciertas limitaciones, la más importante se debe a que si una aplicación no recibe ninguna solicitud en 30 minutos, sus servidores entran en modo suspensión. Esto provoca que la página tarde mucho más tiempo en cargar durante la primera petición, porque se levantan de nuevo los servidores. Esta limitación tiene más impacto en el prototipo desarrollado, ya que si el servidor está en suspensión, será imposible ejecutar la tarea encargada de realizar peticiones periódicas al repositorio Open Data. Por tanto, Heroku no se trata de una opción viable para este proyecto.

Para facilitar el acceso a la aplicación una vez desplegada, es necesario disponer de un nombre de dominio que identifique de manera sencilla al sitio web. Para ello, se ha empleado **Namecheap**[42], que se trata de uno de los registradores de nombres de dominios con alojamiento web más populares. Además, ofrece a los estudiantes paquetes gratuitos para la creación de dominios sin ningún coste.



Figura 2.7: Plataformas para realizar el despliegue

## 2.6. Documentación

La memoria del proyecto ha sido generada empleando  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  con el editor en línea Overleaf.

# Capítulo 3

## Desarrollo de arquitecturas

En esta sección se explica el desarrollo e implementación de las arquitecturas que forman el proyecto. Con ello, se ha creado una página web y una aplicación Android denominadas **DATUR**, su código se encuentra disponible en GitHub, organizado en tres repositorios: backend<sup>1</sup>, frontend web<sup>2</sup> y frontend móvil<sup>3</sup>. En el repositorio para el backend están accesibles los datasets utilizados. Además, también es posible acceder a la página web<sup>4</sup> y descargar la aplicación móvil<sup>5</sup>.

La arquitectura general del proyecto se ha dividido en 3 capas, con la finalidad de gestionar mejor las diferentes responsabilidades. De esta forma se consigue que cada una de las capas se encargue de una serie de tareas específicas. Dentro de la arquitectura también se considera la parte de almacenamiento de datos.

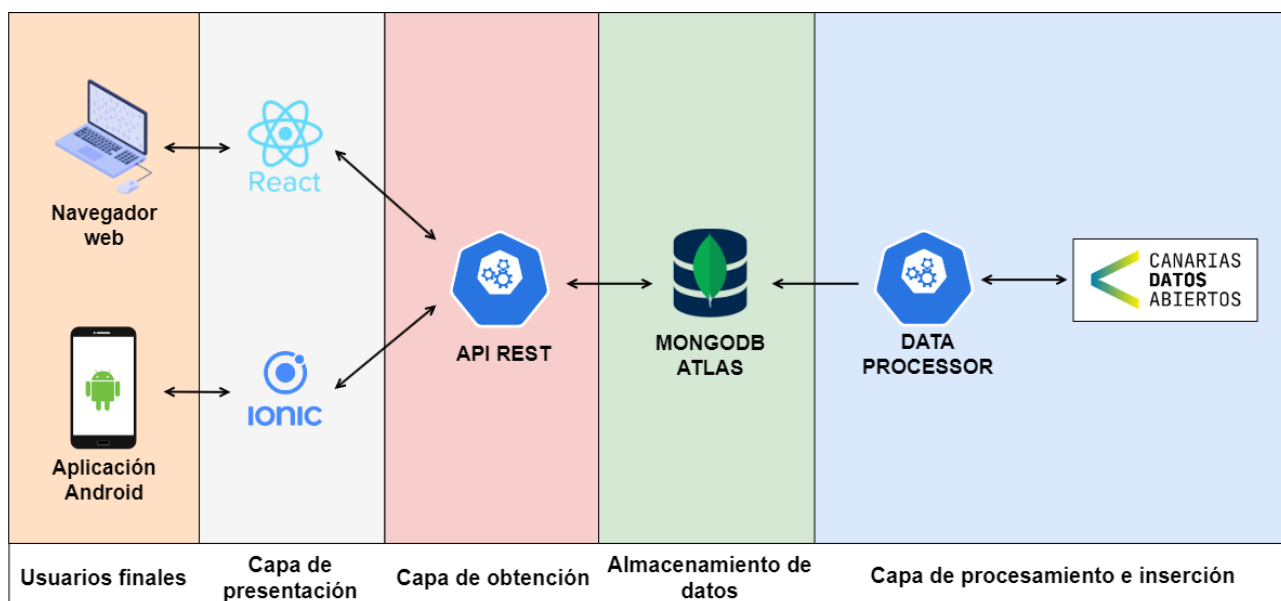


Figura 3.1: Diagrama de la arquitectura general del proyecto

<sup>1</sup> **Repositorio Backend:** <https://github.com/alu0101217741/TFG-DATUR-Backend>

<sup>2</sup> **Repositorio Frontend Web:** <https://github.com/alu0101217741/TFG-DATUR-Frontend-Web>

<sup>3</sup> **Repositorio Frontend Móvil:** <https://github.com/alu0101217741/TFG-DATUR-Frontend-Movil>

<sup>4</sup> **Página web:** <https://datur.me/>

<sup>5</sup> **Aplicación móvil:** [https://drive.google.com/file/d/1DtxZuFrCx\\_D\\_25QPnM9wVWsJlXEwZh6ax/view](https://drive.google.com/file/d/1DtxZuFrCx_D_25QPnM9wVWsJlXEwZh6ax/view)

A continuación, se describe el flujo de este diagrama:

1. El usuario se conecta mediante el navegador web o la aplicación Android. En caso de emplear navegador web se utiliza HTTP, para conectar con el servidor proporcionado por DigitalOcean, donde se hospeda la aplicación Frontend.
2. A medida que el usuario se mueve por las diferentes secciones de la aplicación, se ejecutan peticiones asíncronas a la API REST.
3. En la API REST se enruta la petición en función de su URL, se accede a la base de datos, y se devuelve en formato JSON los datos solicitados.
4. Cuando los datos llegan a la capa de presentación se generan los gráficos, y se renderizan en la pantalla del usuario.
5. De manera independiente al funcionamiento descrito en los pasos anteriores, la capa de procesamiento e inserción realiza de manera periódica peticiones al repositorio Open Data, procesa sus datos y los almacena en la base de datos.

### **3.1. Datos seleccionados**

En primer lugar, se deben seleccionar los datasets del repositorio Canarias Datos Abiertos, que se van a procesar y que permitirán generar las gráficas en la capa de presentación.

En el repositorio se encuentran disponibles para la temática de turismo un total de 930 conjuntos de datos, por ello fue necesario establecer ciertos criterios para seleccionar los que realmente son de interés para el sistema. De esta forma se determinaron cuatro de los aspectos más relevantes del turismo, basados en el número de turistas, el gasto turístico, la estancia de los turistas en Canarias, y las expectativas turísticas. Centrando el estudio en estos cuatro aspectos, el objetivo consistía en emplear datasets que proporcionarían información sobre alguno de ellos, finalmente se han procesado los siguientes conjuntos de datos:

1. Turistas que han visitado Canarias según lugares de residencia por períodos.
2. Turistas principales según lugares de residencia por islas de Canarias y periodos.
3. Estancia media de los viajeros según lugares de residencia por islas de alojamiento de Canarias y periodos.
4. Estancia media según tipos de alojamiento y países de residencia. Islas de Canarias y periodos.
5. Gastos medios, incluyendo desgloses del gasto, según países de residencia por islas de Canarias y periodos.
6. Gasto turístico total según nacionalidades por periodos.
7. Expectativas de la tendencia del grado de ocupación según categorías de los establecimientos por islas de Canarias y periodos.

8. Grado de ocupación previsto para cada mes.
9. Balance de situación, balance de expectativas e índices de confianza hotelera por islas de Canarias y periodos.
10. Expectativas de la marcha del negocio respecto al trimestre anterior según categorías de los establecimientos por islas de Canarias y periodos.
11. Expectativas de los principales factores de la marcha del negocio respecto a otros trimestres según categorías de los establecimiento por islas de Canarias y periodos.

Para consumir estos datasets se emplea la API ofrecida por el repositorio Open Data, la cual está construida mediante la herramienta CKAN.

## 3.2. Backend

El backend del proyecto está formado por las capas de procesamiento e inserción, el almacenamiento de datos y la capa de obtención. En esta sección se describen sus implementaciones y funcionalidades.

### 3.2.1. Capa de procesamiento e inserción

La capa de procesamiento e inserción garantiza la fiabilidad del proyecto, transformando los datos originales a los formatos con los que se trabaja en todo el sistema, y gestionando los posibles errores que se puedan producir. Además, asegura que los datos almacenados, siempre estén actualizados con los últimos datasets disponibles en el repositorio Open Data.

Por tanto, la responsabilidad principal de esta capa consiste en ejecutar los procesos ETL<sup>6</sup>, para lo cual se ha implementado un sistema denominado **Data Processor**:

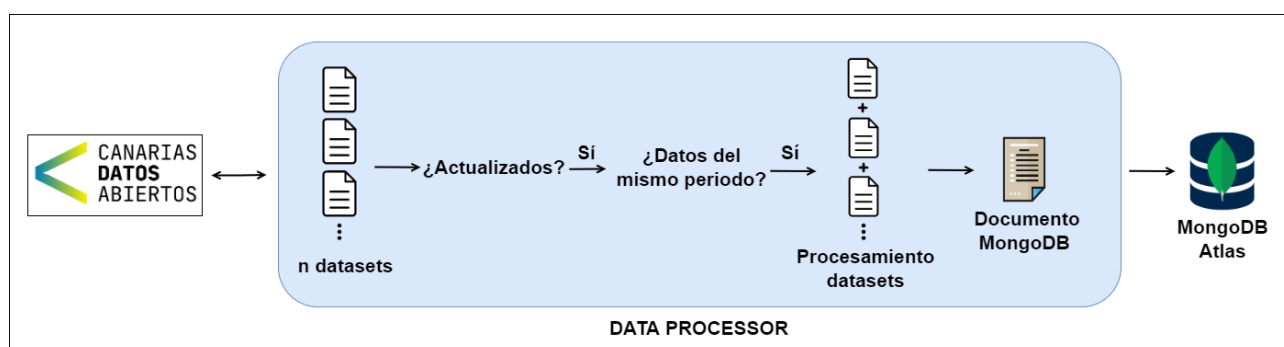


Figura 3.2: Diagrama de la capa de procesamiento e inserción

Es importante tener en cuenta que los documentos de cada una de las colecciones de la base de datos, se construyen procesando un grupo de datasets del repositorio Open Data.

<sup>6</sup>**ETL**: siglas de Extract, Transform and Load (extracción, transformación y carga). Son los procesos que permiten a las organizaciones consumir datos desde múltiples fuentes, formatearlos y cargarlos en otra base de datos para su posterior análisis.

De forma periódica gracias a la utilización de un cron<sup>7</sup> se extraen, del repositorio Canarias Datos Abiertos, el conjunto de datasets que se están utilizando para una colección en concreto.

Tras ello, se comprueba si se han actualizado respecto al anterior procesamiento, en tal caso, se verifica si los datos de todos los datasets pertenecen al mismo periodo de tiempo, ya que puede suceder que algunos tengan datos más actualizados que otros, generando con esto, inconsistencias en el momento de realizar las operaciones de procesamiento.

Si los datasets están sincronizados se procesan y fusionan creando con ello un nuevo documento, que se almacena en la colección correspondiente de la base de datos. Actualmente, en el proyecto se está trabajando con un total de 11 datasets, los cuales se procesan para generar 5 colecciones en la base de datos.

Las clases que implementan el sistema Data Processor se organizan en 3 partes: infraestructura, dominio y aplicación.

## Infraestructura

Se trata de la parte del sistema que interactúa y trabaja directamente con elementos externos, en este caso el repositorio Canarias Datos Abiertos.

Incluye la clase **OpenDataInterface**, que recibe un grupo de identificadores de los datasets que se quieren recuperar del repositorio. Para extraer estos datasets, es necesario hacer dos peticiones para cada uno de ellos. Primero se emplea una de las funciones de la API de CKAN ofrecida por el repositorio, la cual devuelve los metadatos del dataset y las URLs de sus diferentes recursos. Después es necesario realizar otra petición sobre una de estas URLs, para obtener el recurso concreto en formato JSON.

En los formatos JSON que se reciben, existen numerosos campos que no resultan de interés para el posterior procesamiento. Por ello, esta clase también lleva a cabo una limpieza de estos primeros datos, para convertirlos a un formato preparado para iniciar el proceso de transformación. Este formato viene determinado por la clase de dominio llamada **Dataset**, por tanto se crean instancias de esta clase tomando de los datasets solamente aquellos campos que resultan de interés. Los datos que incluyen estas instancias son los que se van a emplear durante la fase de transformación.

**OpenDataInterface** también se encarga de gestionar los posibles errores que puedan suceder en las peticiones, evitando que se propaguen a otras partes del sistema. Estas peticiones al repositorio se realizan empleando Axios.

Las otras clases que forman la infraestructura son aquellas que realizan la transformación de los datos. Actualmente, existen 5 clases de este tipo, una para cada colección de la base de datos.

Todas las clases encargadas de las transformaciones extienden de una clase abstracta denominada **DatasetProcessor**. Esta clase incluye el método **process** que recibe un array con instancias de la clase Dataset, e implementa las siguientes funcionalidades:

---

<sup>7</sup>**Cron:** es un administrador regular de procesos en segundo plano que ejecuta tareas a intervalos regulares de tiempo (por ejemplo, cada minuto, día, semana o mes).

1. Genera un array con el último periodo de actualización de cada uno de los datasets.
2. Comprueba si estos datasets se han actualizado, para ello compara los valores del array generado en el paso anterior, con los periodos de actualización almacenados en memoria desde el anterior procesamiento.
3. Si alguno se ha actualizado, es necesario obtener el último periodo de actualización común para todos ellos. Esto se hace para evitar inconsistencias durante el procesamiento, ya que al fusionar datos de varios datasets, si esto no se tiene en cuenta, puede suceder que si alguno se actualiza, se lleve a cabo el procesamiento, sin disponer de datos para este nuevo periodo en el resto de datasets.
4. Cuando se dispone del periodo común de actualización, se comprueba si es diferente al último guardado, en tal caso se invoca un método abstracto para transformar los datasets. Este método tendrá una implementación particular en cada una de las 5 clases encargadas directamente de la transformación.

Esta implementación permite optimizar en gran medida el rendimiento del sistema, ya que únicamente se realiza el procesamiento si existen nuevos datos. Además, para lograr una optimización aún mayor, en lugar de procesar completamente todos los datasets cada vez que se ejecuten las clases que realizan las transformaciones, solamente se consideran los datos posteriores al último periodo almacenado.

## **Dominio**

Es la parte del sistema donde, a través de la definición de una serie de clases, se establecen los resultados finales que se obtendrán una vez finalizado el procesamiento de los datos.

Por tanto, primero se debe implementar la clase de dominio para definir mediante sus atributos, el formato final que se quiere obtener del procesamiento de un conjunto de datasets. Una vez hecho esto, se tiene que implementar la correspondiente clase en infraestructura, donde se realicen las transformaciones necesarias sobre los datasets, para adaptar los datos al formato establecido por el dominio. Por ello, siempre el resultado devuelto por la clase de infraestructura será una instancia de su respectiva clase de dominio.

Cabe destacar que la utilidad de las clases de dominio, no reside únicamente en definir una serie de atributos, también proporcionan métodos que facilitan las tareas de transformación. Otro aspecto importante consiste en que los atributos de estas clases de dominio, deben estar alineados con los campos que forman los documentos en las colecciones de la base de datos, para evitar errores durante la carga de los datos. En concreto en el dominio se disponen de 5 clases principales.

En cuanto a la clase denominada Dataset, nombrada en la parte de infraestructura, tiene un enfoque diferente al resto de clases de dominio. Su objetivo es servir de contenedor de los campos seleccionados tras la limpieza de los datasets, proporcionando algunos métodos que faciliten la gestión de los mismos.





El periodo que debe transcurrir entre cada ejecución se puede establecer libremente, aunque debe ser mayor al tiempo que supone el procesamiento de todos los datasets involucrados, para evitar problemas de concurrencia. Actualmente se emplea `0 0 20 * * 1`, esto implica que la lógica del sistema Data Processor se ejecuta cada lunes a las 20:00 horas.

### 3.2.2. Capa de obtención

La capa de obtención tiene como responsabilidad actuar como intermediaria entre la base de datos y las aplicaciones del lado del cliente. De esta forma se consigue que el acceso a los datos sea totalmente transparente para el usuario.

En esta capa se encuentra la API, que se trata de un conjunto de definiciones y reglas que establecen cómo las aplicaciones se conectan y comunican entre sí. En concreto se implementa una API REST, la cual es un tipo de API, que aprovecha el protocolo HTTP para permitir la comunicación entre un cliente y una aplicación de servidor, aplicando los principios de diseño de la arquitectura REST<sup>8</sup>[43, 44]:

- **Interfaz uniforme:** todas las solicitudes de API para el mismo recurso deben ser iguales, sin importar la procedencia de la solicitud.
- **Los objetos se manipulan mediante su URI<sup>9</sup>:** el identificador único de los recursos en un sistema REST es la URI y ningún otro elemento.
- **Desacoplamiento del cliente-servidor:** las aplicaciones del lado del cliente y del servidor tienen que ser totalmente independientes. La única información que el lado del cliente debe conocer es el URI del recurso que se solicita.
- **Protocolo cliente/servidor sin estado:** cada petición HTTP incluye toda la información necesaria para ejecutarla, esto hace que ni cliente ni servidor requieran recordar ningún estado previo para procesarla.
- **Capacidad de almacenamiento caché:** siempre que sea posible, los recursos deben poder almacenarse en la memoria caché en el lado del cliente o el servidor.
- **Arquitectura del sistema en capas:** se emplea una arquitectura jerárquica, donde cada capa lleva a cabo una funcionalidad.
- **Código bajo demanda:** temporalmente los servidores pueden ampliar la funcionalidad del cliente mediante el envío de código ejecutable.
- **Las operaciones más importantes relacionadas con los datos son cuatro:** POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).

---

<sup>8</sup>**REST:** siglas de Representational State Transfer, en español Transferencia de Estado Representacional.

<sup>9</sup>**URI:** siglas de Uniform Resource Identifier, en español identificador de recursos uniforme, es una cadena de caracteres que identifica los recursos de una red de forma unívoca.

Para implementar la API REST se emplea el entorno de ejecución Node.js y el framework Express, siguiendo el esquema que se muestra a continuación:

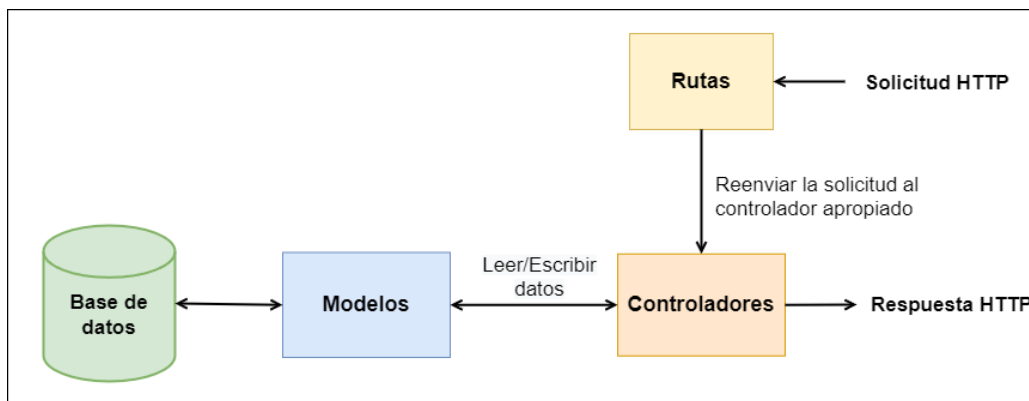


Figura 3.4: Diagrama API REST

De forma general se describen las funciones de cada uno de los elementos que aparecen en el diagrama:

- **Rutas:** reenvían las solicitudes admitidas a las funciones que se encuentran en la capa de los controladores.
- **Controladores:** obtiene los datos solicitados de los modelos de base de datos, y los devuelve al usuario.
- **Modelos:** proporcionan un sencilla interfaz para interactuar con la base de datos.

A continuación, se explica la implementación concreta que se ha realizado:

## Rutas

En primer lugar, es importante establecer que una ruta es un fragmento de código Express que asocia un verbo HTTP (GET, POST, PUT, DELETE, etc.), con una URL y con una función que se llama para manejar la lógica vinculada a la URL. Estas URLs son las que definen finalmente los endpoints<sup>10</sup> de la API REST.

Para definir estas rutas se generan instancias de la clase **express.Router**[45], que permite agrupar un conjunto de rutas y acceder a ellas empleando un prefijo común, implementando su lógica de manera independiente al resto de la API.

Esta implementación evita tener una aplicación Express de gran tamaño formada por multitud de rutas, en su lugar se tendrá una aplicación a la que se incorporan, gracias a las instancias de la clase **express.Router**, otras aplicaciones de menor tamaño formadas por diferentes grupo de rutas.

En el proyecto se dispone actualmente de 5 ficheros, donde se define en cada uno de ellos una instancia de la clase **express.Router**. Cada instancia agrupa las rutas necesarias para interactuar con las 5 colecciones de la base de datos. Únicamente se emplea el verbo HTTP GET, ya que las características del proyecto hacen que el cliente no pueda hacer modificaciones en las colecciones, debido a que esto es responsabilidad de la capa de procesamiento e inserción.

<sup>10</sup>**Endpoint:** URL de un backend que responde a una petición, no está diseñado para interactuar con el usuario final, sino para proporcionar la información que se debe utilizar o renderizar desde el frontend.

## Controladores

Las funciones encargadas de gestionar la lógica asociada a las rutas, se definen en otros ficheros denominados controladores. Esto permite mantener un código más limpio y legible.

En el proyecto estas funciones se encargan de acceder a las colecciones de base de datos a través de los modelos, y devolver al cliente el recurso que ha solicitado, junto con el código HTTP[46] correspondiente.

## Modelos

Los modelos permiten interactuar con las diferentes colecciones definidas en la base de datos. Para lograr esto se emplea el ODM mongoose que actúa como interfaz de MongoDB.

Primero se definen los esquemas mediante la interfaz **Schema** de mongoose, para establecer los campos que se van a almacenar en cada documento, junto con sus requisitos de validación y valores predeterminados.

Posteriormente, estos esquemas se compilan en modelos empleando el método **mongoose.model()**, cada modelo se asocia a una colección de documentos en la base de datos MongoDB. De esta forma, empleando los modelos es posible buscar, crear, actualizar y eliminar documentos en la colección correspondiente.

Al existir 5 colecciones en la base de datos, se definen la misma cantidad de modelos para poder interactuar con cada una de ellas.

## Fichero principal

Por último, para que funcione la API REST es necesario un fichero principal que incluya el código para arrancar el servidor. Aquí es donde se importa Express y se definen sus middlewares.

Los middlewares son funciones que se ejecutan después de que el servidor recibe la solicitud y antes de que se envíe la respuesta. En concreto se definen 2 tipos de middlewares:

- **Middleware de terceros:** son módulos Node.js que se incluyen para incorporar funcionalidades adicionales, estas se añaden a nivel de aplicación, de forma que se ejecutan cada vez que se recibe una solicitud. Los middleware de terceros utilizados son:
  - **morgan**[47]: middleware para Node.js que registra solicitudes y errores HTTP, de esta forma se simplifica el desarrollo.
  - **cors**[48]: paquete Node.js que proporciona un middleware Express que puede ser usado para habilitar CORS<sup>11</sup> con varias opciones. En este caso se habilitan todas las solicitudes CORS.

---

<sup>11</sup>**CORS:** Cross-Origin Resource Sharing, se trata de un mecanismo que utiliza cabeceras HTTP adicionales, para permitir que un agente de usuario obtenga permisos para acceder a recursos seleccionados desde un servidor, en un origen distinto al que pertenece.

- **Middleware a nivel de enrutador:** permiten que una aplicación se divida en varias piezas, logrando una mayor modularidad y menor acoplamiento. Esto se consigue vinculando cada middleware a una instancia de la clase `express.Router()`, lo que hace posible incluir en la aplicación Express cada uno de los grupos de rutas que hemos definido anteriormente de manera independiente.

Por tanto, al llegar una petición a la API REST primero se ejecutarán los middlewares de terceros, tras ello, del conjunto de middlewares a nivel enrutador, se ejecutará la ruta que coincida con la URL de la solicitud.

### 3.2.3. Almacenamiento de datos

La base de datos utilizada en el proyecto es MongoDB, en concreto se utiliza su implementación en la nube a través de MongoDB Atlas.

Actualmente se disponen de 5 colecciones en la base de datos, cuyos documentos se generan mediante el procesamiento de 11 datasets del repositorio Open Data. A continuación, se muestra la relación existente entre los datasets originales y la colección donde se almacenan los documentos obtenidos tras el procesamiento.

<b>Colección MongoDB</b>	<b>Datasets de Canarias Datos Abiertos</b>
Número de turistas	Turistas que han visitado Canarias según lugares de residencia por períodos Turistas principales según lugares de residencia por islas de Canarias y períodos
Gasto turístico	Gastos medios, incluyendo desgloses del gasto, según países de residencia por islas de Canarias y períodos Gasto turístico total según nacionalidades por períodos
Estancia turística	Estancia media de los viajeros según lugares de residencia por islas de alojamiento de Canarias y períodos Estancia media según tipos de alojamiento y países de residencia. Islas de Canarias y períodos
Expectativas tasa de ocupación	Expectativas de la tendencia del grado de ocupación según categorías de los establecimientos por islas de Canarias y períodos Grado de ocupación previsto para cada mes
Expectativas progreso del negocio	Balance de situación, balance de expectativas e índices de confianza hotelera por islas de Canarias y períodos Expectativas de la marcha del negocio respecto al trimestre anterior según categorías de los establecimientos por islas de Canarias y períodos Expectativas de los principales factores de la marcha del negocio respecto a otros trimestres según categorías de los establecimiento por islas de Canarias y períodos

Tabla 3.1: Relación colecciones base de datos y datasets del repositorio Open Data

## 3.3. Frontend

El frontend de la aplicación web y de la aplicación móvil se encuentran dentro de la capa de presentación en la arquitectura general del proyecto.

### 3.3.1. Frontend Web

Como ya se ha comentado, para el desarrollo del frontend de la aplicación web se ha empleado React, React-Bootstrap, Highcharts y CSS.

Además, cabe destacar que es una **Single-Page application** (SPA), es decir, se trata de una aplicación donde todo su contenido se carga en un solo documento web. Por tanto, todos los códigos HTML, JavaScript y CSS se cargan una sola vez, esto provoca que al navegar por las diferentes secciones de la aplicación, el contenido ya está precargado de antemano. Algunos recursos pueden ser cargados dinámicamente si se requieren, pero no tendrá que cargar toda la página nuevamente, solo los nuevos elementos. Esto aporta a los usuarios una experiencia más fluida y rápida. Para construir la SPA se ha utilizado React junto con la librería React Router[49], que proporciona funcionalidades de enrutamiento para este tipo de aplicaciones.

También es importante tener en cuenta que la web sigue un **diseño responsivo**, por lo que se visualiza correctamente en diferentes tamaños de pantallas.

Con estas tecnologías se han desarrollado las siguientes pantallas:

#### Página de inicio

Esta es la primera pantalla que aparece cuando se accede a la aplicación web. En la parte superior se incluye el nombre de la aplicación, junto con su eslogan y una pequeña descripción sobre el contenido que se puede encontrar. Esto hace que rápidamente se conozca la finalidad de la aplicación.

Tras el eslogan, se muestra un vídeo con imágenes que reflejan aspectos relevantes para el turismo de Canarias.



Figura 3.5: Página de inicio aplicación web

Tras ello, se muestran 4 secciones: número de turistas, gasto turístico, estancia turística y expectativas turísticas. Se ha decidido establecer estas secciones ya que cubren muchos aspectos relevantes relacionados con el turismo en Canarias, además, permiten agrupar correctamente las 5 colecciones disponibles en la base de datos.

En cada una de las tarjetas que muestran las secciones, se incluye su título, descripción, fuente de la que se obtienen los datos, y la actualización estimada de los datasets que se utilizan para crear las gráficas en esa sección. Al pulsar sobre las tarjetas, llevan a la pantalla donde se renderizan las visualizaciones de datos correspondientes.

## Nuestras secciones



### Número de turistas

Conoce el número de turistas que visitan las Islas Canarias, incluyendo su evolución, nacionalidades y distribución por islas.

*Actualización estimada: mensual*



### Gasto turístico

Observa la evolución del gasto turístico por año y trimestre, con un desglose del gasto por conceptos.

*Actualización estimada: mensual*



### Estancia turística

Infórmate sobre la estancia media de los turistas, teniendo en cuenta el año, lugar de residencia, isla y tipo de alojamiento.

*Actualización estimada: mensual*



### Expectativas turísticas

Consigue una visión del futuro del sector turístico, a través de las expectativas del grado de ocupación y la marcha del negocio.

*Actualización estimada: mensual*

Figura 3.6: Secciones aplicación web

Finalmente, se incluye un pie de página con una descripción de la aplicación web, enlaces recomendados e información de contacto.



Figura 3.7: Pie de página web

## Visualizaciones

En la aplicación web están disponibles 4 secciones, navegando por todas ellas se pueden observar un total de 15 gráficos con diferentes estilos, opciones configurables y maneras de visualizar la información muy diversas. Por ello, únicamente se mostrará la sección **Número de turistas**, para que sirva como ejemplo del formato seguido para crear las secciones. Si se desea navegar por la página web, y visualizar todo su contenido, está disponible a través del enlace <https://datur.me/>.

Al acceder a una sección, la parte superior de la web cambia, para incluir un menú que facilita la navegación entre los distintos apartados del sitio. Siempre es posible regresar a la página de inicio, pulsando sobre el nombre de la aplicación.

En cuanto al contenido, lo primero que se muestra es un apartado que informa acerca de la finalidad de la sección, incluyendo enlaces a los datasets que se utilizan para generar las gráficas en esta parte de la aplicación.



Figura 3.8: Sección Número de turistas

Tras el apartado informativo, comienzan las visualizaciones de los datos, cabe destacar que la mayoría de los gráficos poseen una alta interactividad, ya que se ofrece al usuario un conjunto de botones donde puede realizar varias acciones: cambiar el tipo de gráfico, el año de los datos, su organización temporal (anual o trimestral), etc. También se incluyen opciones en cada gráfico que permiten ampliarlo en pantalla completa, imprimirlo, y descargarlo en formatos PDF, PNG, JPEG y SVG.

A continuación, se muestran las visualizaciones que componen la sección Número de turistas.

## Visualización de los datos

### Evolución del número de turistas

Se representa la evolución que ha experimentado la cifra anual de turistas que visitan las Islas Canarias.

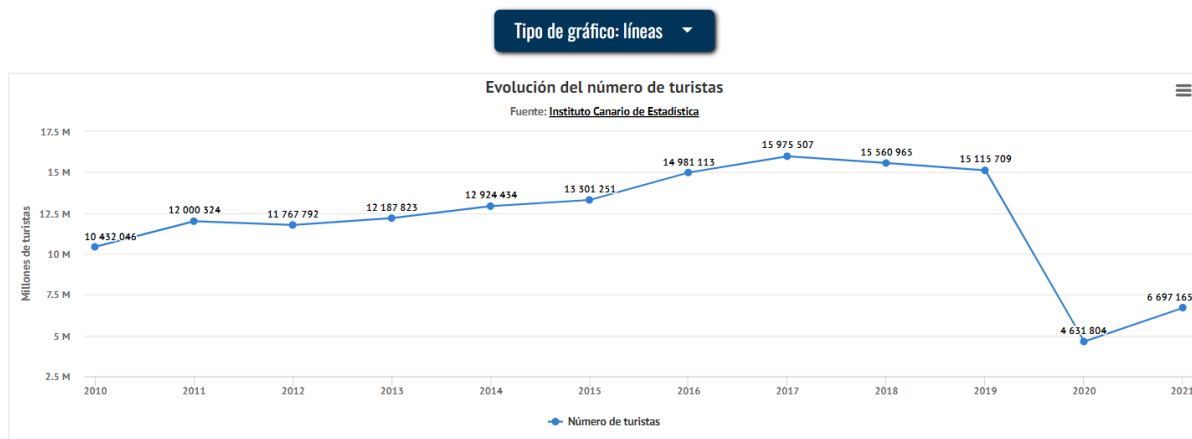


Figura 3.9: Gráfica evolución número de turistas

### Nacionalidades que visitan Canarias

En esta gráfica se muestran las principales nacionalidades que visitan Canarias, se ordenan según el número de turistas que llegan al archipiélago procedentes de ese país de origen. Para la consulta se encuentran disponibles los últimos cuatro años de los que se tienen datos, y se ofrece la posibilidad de organizar la información por año, lo que permite conocer el número total de turistas de cada nacionalidad, o por trimestre, lo que hace posible saber la distribución de los turistas a lo largo del año.



Figura 3.10: Gráfica nacionalidades que visitan Canarias



## Distribución de los turistas

Se muestra el número de turistas que visitan cada una de las Islas Canarias, están disponibles los últimos cuatro de los que se tienen datos. Cabe destacar que se agrupan la cantidad de turistas que han visitado La Gomera y el Hierro, ya que este valor se ha estimado a través de la información del resto de islas y el total de turistas llegados a Canarias. Se ha llevado a cabo esta estimación debido a la falta de información para estas dos islas en la fuente oficial.



Figura 3.11: Gráfica distribución de los turistas

### 3.3.2. Frontend Móvil

El desarrollo del frontend para la aplicación Android se ha llevado a cabo empleando principalmente Ionic React, TypeScript, Highcharts y CSS. Cabe destacar que una vez finalizado el desarrollo, para construir la aplicación y poder utilizarla en dispositivos móviles, se ha utilizado el IDE Android Studio.

En cuanto al diseño es muy similar a la página web, con ciertos ajustes para adaptarlo al menor tamaño de las pantallas. Estos ajustes consisten en cambiar el vídeo de la página de inicio por una imagen, incluir un menú desplegable, añadir una sección *Sobre nosotros* que incorpora la información disponible en el pie de página, y adaptar botones y gráficas a las características móviles.



Figura 3.12: Pantalla de inicio y menú desplegable en móvil

A continuación, se muestra algunas de las gráficas de la sección **Número de turistas**.

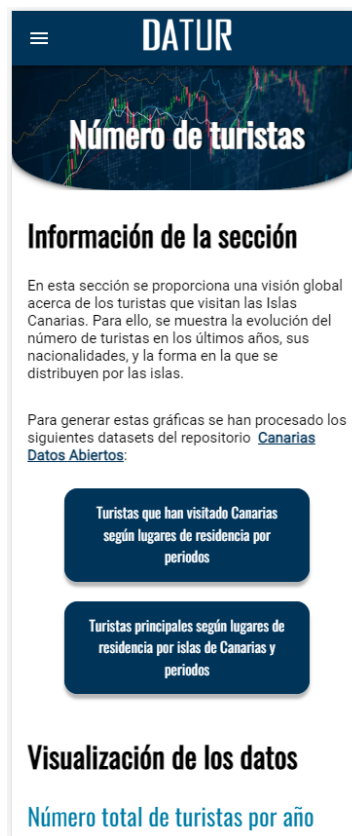
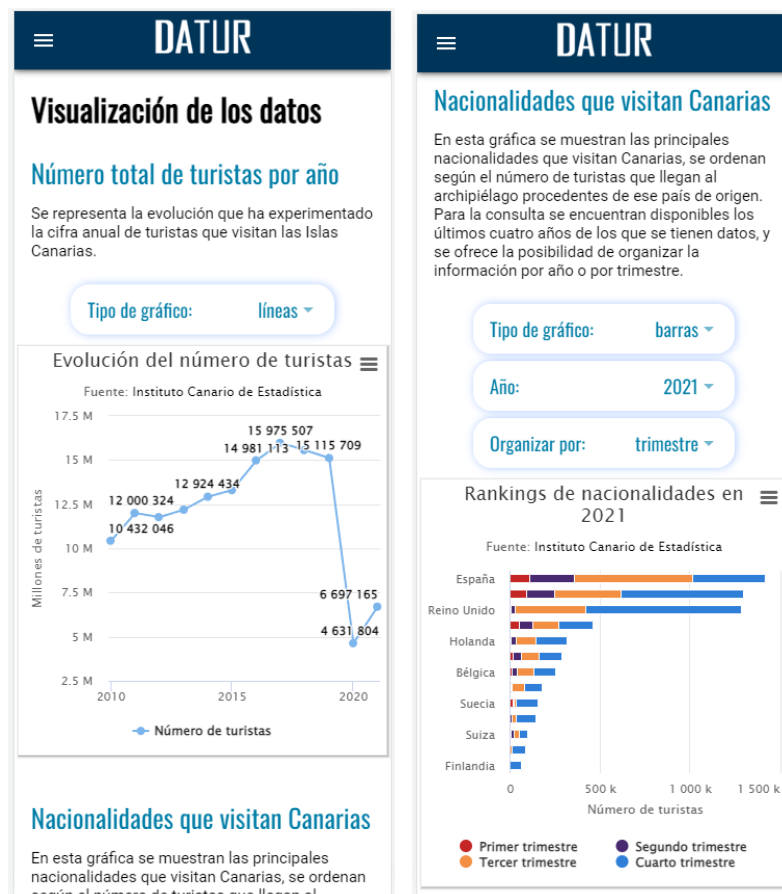


Figura 3.13: Sección Número de turistas en móvil



(a)

(b)

Figura 3.14: Visualización de gráficas en móvil

### 3.4. Despliegue

Una vez desarrollada la aplicación web se debe desplegar, para conseguir que el sistema pueda ser utilizado por los usuarios. Esto se ha realizado mediante DigitalOcean a través de la creación de un droplet, que es el nombre utilizado por la compañía para designar una máquina virtual basada en Linux ejecutada sobre hardware virtualizado. Las opciones de configuración que se han establecido para crear el droplet son las siguientes:

- Distribución de Linux: Ubuntu 22.04 x64
- 1 vCPU
- 1 GB de memoria RAM
- 10 GB de almacenamiento SSD
- 500 GB de transferencia de datos

Tras realizar las configuraciones necesarias en el droplet, debemos clonar en el servidor los repositorios de GitHub que contienen el código del backend y del frontend web. Después de instalar las dependencias de ambos proyectos, se pueden comenzar a ejecutar, para ello tenemos que utilizar **pm2**[50], que se trata de un gestor avanzado de procesos de producción para Node.js, que permitirá mantener siempre en ejecución tanto el backend

como el frontend. En concreto la API REST se está ejecutando en el puerto 3000, por su parte el frontend se ejecuta en el puerto 3001. El sistema Data Processor se ejecuta internamente en el servidor, por lo que no expone ningún puerto.

Para gestionar las peticiones entre los clientes y el servidor se ha empleado **NGINX**[51], que se trata de un balanceador de carga, servidor web y proxy inverso de alto rendimiento y de código abierto. Por tanto, ofrece el contenido estático de un sitio web de forma rápida y con un bajo consumo de recursos. Además, es fácilmente configurable, y permite miles de conexiones simultáneas. Todo ello aporta mayor velocidad y escalabilidad.

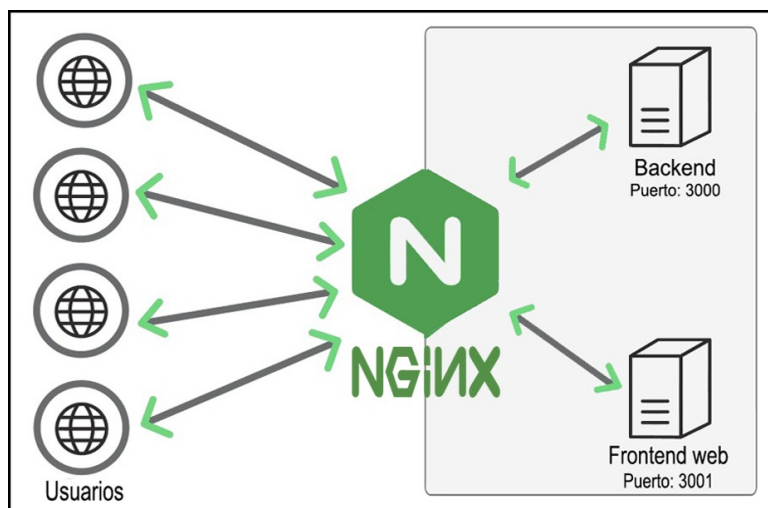


Figura 3.15: Diagrama sistema desplegado

En este momento para acceder a la aplicación web, es necesario formar la URL incluyendo la dirección IP del servidor. Para evitar esto y facilitar el acceso al sistema debemos emplear un nombre de dominio, este se ha adquirido en el registrador de nombres de dominio Namecheap, a través de sus planes gratuitos para estudiantes. El dominio adquirido ha sido **datur.me**. Tras realizar las configuraciones necesarias, es posible acceder a la aplicación web a través de la URL <https://datur.me/>.

Por último, se ha incorporado SSL<sup>12</sup> en la aplicación web mediante un registro gratuito en Let's Encrypt[52], que se trata de una autoridad de certificación sin ánimo de lucro que ofrece un bot llamado Certbot para añadir SSL automáticamente. Una vez instalado el bot en el servidor, se encarga de validar el sitio web, de forma que el dominio del sistema posee SSL.

<sup>12</sup>**SSL**: *Secure Sockets Layer*, es la tecnología estándar para mantener segura una conexión a Internet, protegiendo cualquier información confidencial entre dos sistemas, con ello se evita que se pueda leer o modificar cualquier dato que se transfiera.

# Capítulo 4

## Estudio de viabilidad económica

### 4.1. Funcionalidades adicionales

El prototipo de aplicación desarrollado cubre con los objetivos establecidos para el proyecto, y brinda la posibilidad de incrementar la digitalización y el conocimiento del sector turístico canario. Sin embargo, debido a las limitaciones de tiempo y recursos, existen algunas funcionalidades adicionales que no se han podido añadir, pero que resultarían interesantes para una aplicación profesional de visualización de datos en tiempo real sobre el sector turístico, donde no existieran las limitaciones de este proyecto.

Tras analizar las soluciones que ofrece el mercado, se ha concluido la necesidad de incorporar las siguientes funcionalidades en la aplicación profesional, con el objetivo de destacar sobre la competencia:

- **Registro de usuario:** implementar un sistema de registro permite establecer los perfiles de seguridad, con ello es posible determinar las opciones y datos a los que puede acceder cada usuario, de acuerdo al plan de suscripción que tenga contratado. Además, es un mecanismo para obtener datos, lo que aporta un mayor conocimiento acerca de los individuos y organizaciones que utilizan la aplicación.
- **Notificaciones de actualización:** al tratarse de una aplicación que actualiza su contenido a medida que lo hace la fuente oficial de la que se extraen los datos, una vez que el usuario ha consultado la información disponible, el siguiente momento de mayor interés será cuando existan nuevos datos. Esto obliga al usuario a revisar constantemente la aplicación para detectar posibles actualizaciones. La solución planteada consiste en implementar un sistema, que notifique al usuario por correo electrónico cuando se produzca alguna actualización. Además, para no enviar una gran cantidad de mensajes que puedan ser molestos para el usuario, se habilitará la posibilidad de que elija aquellas secciones de las cuales quiere recibir notificaciones.
- **Dashboard personalizados:** los usuarios deben poder seleccionar aquellas gráficas que más les interesan, de acuerdo a los datos y métricas que les resulten más relevantes para la toma de decisiones en sus negocios. De esta forma es posible crear un panel personalizado, que incluye las gráficas que son de interés directo para el usuario, permitiendo que todas ellas puedan ser visualizadas rápidamente con un simple vistazo. Además, evita que el usuario tenga que recorrer todas las secciones para observar la gráfica concreta que desea.

- **Generación de informes:** en el prototipo diseñado se puede exportar de forma individual cada una de las gráficas en múltiples formatos. En la aplicación profesional se debe mantener esta funcionalidad, incorporando la opción de generar informes que engloben toda la información disponible en una sección.
- **Creación de visualizaciones:** se debe incluir un sistema que permita a los usuarios subir archivos de datos en determinados formatos y con un esquema preestablecido, para generar a partir de ellos visualizaciones estándar.
- **Aumentar fuentes de datos y datasets procesados:** es necesario incorporar nuevas fuentes de datos, incluyendo otros repositorios Open Data que proporcionen información relevante sobre el turismo autonómico, nacional e internacional. Además, también hay que incrementar la cantidad de datasets procesados, con el objetivo de ampliar la información que ofrece la aplicación.

## 4.2. Desarrollo del proyecto

En esta sección se pretende responder a las siguientes preguntas: ¿Cuál es el coste total del proyecto? ¿Cuánto tiempo se necesitaría para completarlo? ¿Cuándo se recuperaría la inversión inicial?

Para responderlas se ha simulado el desarrollo profesional del proyecto empleando el software **Project Libre**<sup>1</sup>, donde se han definido las tareas necesarias, y se han establecido sus duraciones y costes. Con ello ha sido posible estimar la duración del proyecto, su coste total y su camino crítico.

El proyecto se ha dividido en 4 fases:

- **Fase de Análisis:** esta primera fase tiene gran importancia, ya que implica la comprensión del problema que se trata de resolver, y establece las bases del software que se va a desarrollar. Para ello se especifican detalladamente en un documento los requisitos y funcionalidades que debe tener el producto final, con lo que es posible establecer plazos de entrega y costes. Además, este documento debe estar cerrado, evitando incluir nuevos requisitos a mitad del desarrollo. Cabe destacar que la comunicación es fundamental en esta fase, de forma que se realizan reuniones con el equipo involucrado, para asegurar que los avances en el análisis son adecuados.
- **Fase de Diseño:** en esta segunda fase se estudian las posibles implementaciones para el software que se desea construir, y se establece su arquitectura general. Por tanto, consiste en diseñar una solución que satisfaga los requisitos definidos en la etapa anterior. Para ello, se estudian y seleccionan las fuentes de datos, y las tecnologías que mejor se adaptan al proyecto. Una vez hecho esto, se diseñan las arquitecturas que constituirán el producto final. También se realiza una aproximación a la estructura de la base de datos, y se establece el diseño de las interfaces de usuario para la aplicación web y la aplicación móvil.

---

<sup>1</sup>**Project Libre:** software de administración y gestión de proyectos de código abierto.

- Fase de Desarrollo:** se comienzan a desarrollar las tareas y se construye la aplicación, cumpliendo con los objetivos establecidos en el proyecto. Para ello, se estudian los datasets disponibles en las fuentes de datos, y se seleccionan los más adecuados. Una vez hecho esto, se implementan los módulos encargados de procesar los datasets, se desarrolla la base de datos para almacenar los datos procesados, y finalmente se lleva a cabo la API REST que permita el acceso a los mismos. En cuanto al frontend, se implementan las interfaces de la web y la aplicación móvil. Durante todo el proceso se realizan los test necesarios para verificar el software que se va desarrollando. Al finalizar esta fase, se comprueba el correcto funcionamiento del sistema en su conjunto (backend y frontend), para asegurar que el producto cumple con las funcionalidades solicitadas y los estándares de calidad.
- Fase de Despliegue:** en la última fase del proyecto se realizan las actividades para que las aplicaciones desarrolladas estén disponibles en el mercado para su uso. Esto implica planificar los entornos en los que el sistema debe funcionar, considerando tanto hardware como software: configuración física de equipos, redes de interconexión, accesos a sistemas externos, subida de la aplicación móvil a las plataformas correspondientes, etc. Tras el despliegue se deben realizar pruebas para comprobar el correcto funcionamiento en los nuevos entornos, así como evaluar el rendimiento en los mismos.

Al finalizar las 4 etapas, el proyecto comienza a generar ingresos, pero también a partir de este punto se debe tener en cuenta los gastos de producción, debidos al mantenimiento de los entornos donde está desplegado y de la propia aplicación (actualizaciones que corrijan errores, necesidad de nuevas funcionalidades, ...).

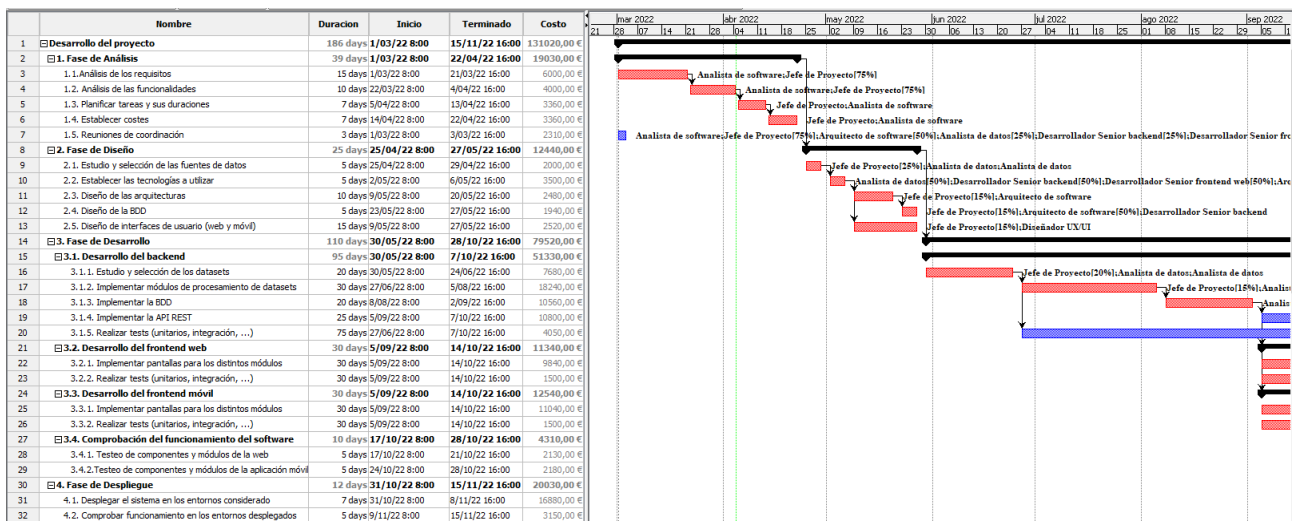


Figura 4.1: Diagrama de Gantt (1)

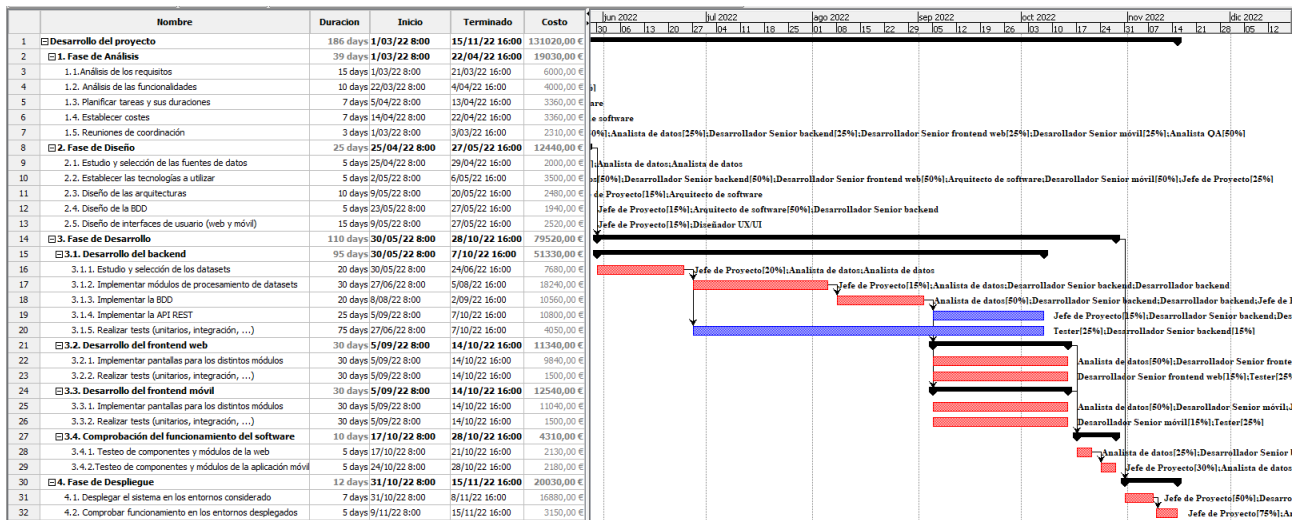


Figura 4.2: Diagrama de Gantt (2)

Si se desea observar mejor el diagrama de Gantt, el proyecto creado mediante el software Project Libre está disponible para su descarga y análisis<sup>2</sup>.

Cabe destacar que durante la fase de análisis se ha considerado que será necesario dedicar 4 horas semanales, para realizar reuniones de coordinación entre los miembros del equipo. Por ello, en el Project Libre, se asigna a esta tarea una duración de 3 días, ya que representa un total de 24 horas laborales, que se distribuyen a razón de 4 horas durante las 6 semanas de análisis. Otro aspecto relevante, es que en la fase de despliegue se ha tenido que incorporar un coste adicional, debido a la importante cantidad que supone. Este coste es derivado de adquirir un nombre de dominio para el sitio web, tras estudiar en el registrador Namecheap, el coste de adquirir y utilizar el dominio datur.com, se ha concluido que su adquisición supondría en torno a 11000 €.

Para llevar a cabo el proyecto se ha definido un equipo formado por los siguientes perfiles:

- **Jefe de proyecto:** es el responsable de que el proyecto se realice dentro de los plazos y presupuesto establecidos. Posee un gran conocimiento del producto que se pretende construir, y participa durante todo el proceso de desarrollo, gestionando los problemas que puedan surgir y supervisando los avances.
- **Analista de software:** se encarga de comprender y determinar las funcionalidades y requisitos del producto que se quiere obtener, así como de asegurar que la solución final que se va a desarrollar cumple con estas necesidades.
- **Arquitecto de software:** crea una solución técnica que permite cumplir con los requisitos del producto, para ello establece la estructura general que tendrá el sistema.
- **Analista de datos:** se encarga de transformar los datos en información que mejore la toma de decisiones. Además, es el responsable de garantizar que todos los involucrados en el proyecto entiendan los datos con los que trabajan. Por tanto,

<sup>2</sup>Proyecto desarrollado en Project Libre: <https://drive.google.com/file/d/16uopwbZ3r59ZIrVVI3CDtM9KDNIt9cPy/view?usp=sharing>



este perfil es fundamental en el momento de seleccionar las fuentes de datos y los datasets a utilizar. También resulta interesante su apoyo en todas aquellas tareas relacionadas directamente con los datos.

- **Diseñador UX/UI:** diseñan las interfaces web y móvil teniendo en cuenta la forma en la que los usuarios interactúan con ellas.
- **Desarrolladores:** son los encargados de implementar las arquitecturas que pongan en funcionamiento el producto. Serán necesarios distintos profesionales que cubran las diferentes partes del sistema: backend, frontend web y frontend móvil.
- **Tester:** a medida que se desarrolla el software, son los responsables de planificar y realizar las pruebas necesarias, para comprobar el correcto funcionamiento del código implementado.
- **Analista Quality Assurance (QA):** asegura el correcto funcionamiento del producto final, y verifica que cumpla con ciertos criterios de calidad (fácil de usar, intuitivo, sin errores, optimizado, etc).

	Nombre	Tipo	Unidades Max	Tasa Estandar
1	Jefe de Proyecto	Trabajo	100%	40,00 €/hora
2	Analista de software	Trabajo	100%	20,00 €/hora
3	Arquitecto de software	Trabajo	100%	25,00 €/hora
4	Analista de datos	Trabajo	100%	20,00 €/hora
5	Analista de datos	Trabajo	100%	20,00 €/hora
6	Diseñador UX/UI	Trabajo	100%	15,00 €/hora
7	Desarrollador Senior backend	Trabajo	100%	30,00 €/hora
8	Desarrollador backend	Trabajo	100%	20,00 €/hora
9	Desarrollador Senior frontend web	Trabajo	100%	25,00 €/hora
10	Desarrollador Senior móvil	Trabajo	100%	30,00 €/hora
11	Tester	Trabajo	100%	15,00 €/hora
12	Analista QA	Trabajo	100%	15,00 €/hora

Figura 4.3: Recursos humanos del proyecto y sus salarios

Tras definir todas las tareas necesarias y establecer sus duraciones, se obtiene un tiempo estimado para completar el proyecto de **186 días laborales**, comenzando el **01 de marzo de 2022** y finalizando el **15 de noviembre de 2022**, con una jornada laboral de 8 horas diarias de lunes a viernes. Por último, el costo total de llevar a cabo el proyecto es de **131.020 €**.

### 4.3. Modelo de Comercialización

Una vez finalizado el desarrollo del proyecto, el producto final estará disponible en el mercado, por ello es necesario diseñar un modelo de comercialización que permita el retorno de la inversión inicial y la obtención de futuros beneficios.

El modelo de comercialización se basa en tres planes de suscripción mensuales:

- **Standard:** 5 € por usuario/mes
- **Pro:** 15 € por usuario/mes
- **Premium:** 25 € por usuario/mes

Cabe destacar que el acceso a la aplicación web será gratuito, sin embargo, sólo será visible una pequeña parte del contenido de la misma. Dentro de este contenido tendrá mayor protagonismo, aquellas secciones donde se explique lo que ofrece el producto y sus diferentes planes de suscripción.

	<b>Standard</b>	<b>Pro</b>	<b>Premium</b>
Número de secciones <sup>3</sup>	5	10	Ilimitado
Exportación de gráficas en múltiples formatos	Sí	Sí	Sí
Generación de informes	No	Sí	Sí
Dashboard personalizados	No	Sí	Sí
Acceso aplicación iOS y Android	No	Sí	Sí
Creación de visualizaciones	No	No	Sí
Notificaciones de actualización	No	No	Sí
Acceso a los datos procesados <sup>4</sup>	No	No	Sí

Tabla 4.1: Comparación planes de suscripción

Además de los ingresos, también se deben tener en cuenta los gastos que implica tener el producto disponible en el mercado.

En primer lugar, hay que considerar el gasto del servidor donde esté alojada la aplicación, para contratarlo se ha elegido la plataforma DigitalOcean, al presentar servicios que cumplen perfectamente con los requisitos de una aplicación profesional.

Entre todos los planes que ofrece[53, 54], se ha decidido emplear el plan *Memoria Optimizada* para crear el droplet, ya que proporciona CPU dedicada, y con la memoria RAM adicional se evitan problemas de cambios excesivos en el disco o errores de falta de memoria, mejorando el rendimiento y estabilidad de la aplicación. Además, es el plan recomendado para aplicaciones de procesamiento de grandes conjuntos de datos en tiempo real. Dentro de este plan, las configuraciones iniciales establecidas para el droplet son:

- 16 GB de memoria RAM

<sup>3</sup>Durante la contratación del plan se ofrece un conjunto de secciones, cada una de las cuales cubre un aspecto relevante del turismo (economía, alojamiento, expectativas,...), el cliente debe seleccionar de entre todas las secciones cuál desea tener disponible en su plan.

<sup>4</sup>Se proporciona al cliente la posibilidad de consumir directamente los datos que se obtienen del procesamiento, esto se hará permitiendo su acceso a la API REST, con ello podrá reutilizar estos datos procesados en sus propios proyectos.

- 50 GB de almacenamiento SSD
- 2 vCPUs
- 4 TB de transferencia de datos

DigitalOcean factura el coste por cada hora de uso, en el caso de la configuración elegida el precio es de **0.125 €/hora**, lo que supone un coste mensual estimado de **84 €**.

Es importante tener en cuenta que a medida que incrementen el número de usuarios de la aplicación, los recursos del servidor también deben aumentar, para continuar proporcionando un servicio adecuado. Por ello, se considera que tras el primer año en el mercado, se debe mejorar la configuración del servidor, para que disponga de los siguientes recursos:

- 32 GB de memoria RAM
- 100 GB de almacenamiento SSD
- 4 vCPUs
- 6 TB de transferencia de datos

Esta mejora implica que el precio sea de **0.25 €/hora**, lo que supone un coste mensual estimado de **168 €**.

Otros costes a tener en cuenta son aquellos derivados del nombre de dominio. Durante la simulación del desarrollo profesional del proyecto, ya se considero la adquisición del mismo, pero disponer de un dominio también implica un coste periódico. Tras estudiar en el registrador Namecheap, el coste de utilizar el dominio datur.com, se concluye que tendría una tarifa de registro anual de **15 €/año**.

También es importante la seguridad de la página web, por ello se debe disponer de un certificado SSL, para encriptar toda la información relacionada con ella, y protegerla frente a terceros. Además, este certificado es relevante en términos de posicionamiento, ya que los sitios sin SSL tienden a ser desplazados hacia abajo en las clasificaciones de búsqueda. Namecheap también ofrece una gama completa de certificados SSL, pudiendo adquirir el certificado por un coste aproximado de **50 €/año**.

Por último, se debe considerar un gasto de mantenimiento mensual, ya que es posible que ciertas personas del equipo tengan que resolver problemas que puedan surgir, o revisar que el sistema funciona correctamente. Se estima para ello un total de **400 €/mes**.

Una vez se dispone de los planes de suscripción y de los costes que implica tener el producto en el mercado, se ha realizado un estudio que estima las contrataciones de los diferentes planes para las 100 primeras semanas. Con el estudio realizado<sup>5</sup>, se ha concluido que el proyecto alcanzaría su punto **ROI** tras **89 semanas** en el mercado, es decir, serían necesarios 20 meses y medio para recuperar la inversión inicial. A partir de este punto, generaría beneficios para sus inversores.

---

<sup>5</sup>El estudio realizado se puede encontrar en: [https://docs.google.com/spreadsheets/d/1BK-fCz930okvtmP\\_gl30AeQ1SoEVFDpNZdTtwiw-jfw/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1BK-fCz930okvtmP_gl30AeQ1SoEVFDpNZdTtwiw-jfw/edit?usp=sharing)



Figura 4.4: Gráfico que refleja punto ROI

# Capítulo 5

## Conclusiones y líneas futuras

En conclusión, la gran cantidad de datos que se encuentran disponibles para su reutilización a través de plataformas Open Data, ofrecen la posibilidad de crear aplicaciones innovadoras, que utilicen el procesamiento y la visualización como principales herramientas para generar información de valor. Logrando con ello, mayor conocimiento del sector donde se apliquen, en este caso el turismo, lo que implica ventajas competitivas y mejor toma de decisiones.

A lo largo de este proyecto se ha realizado un estudio de mercado, con el objetivo de analizar las soluciones que actualmente ofrecen procesamiento de datos turísticos. También, se han seleccionado las mejores tecnologías y arquitecturas en función de las características del proyecto. Con ello, se ha desarrollado un prototipo de aplicación web y aplicación móvil totalmente funcionales, que cumplen con las expectativas que se tenían al inicio, proporcionando visualizaciones de diversos estilos que siempre emplean datos actualizados. Esto ha permitido crear una herramienta que proporciona un análisis detallado de cuatro aspectos muy relevantes para el sector turístico canario. Además, se ha demostrado la viabilidad económica de llevar a cabo un proyecto de estas características. Por tanto, se han cumplido con todos los objetivos que se establecieron al principio de este Trabajo Fin de Grado.

Por último, como líneas futuras sería importante incluir en la aplicación, las funcionalidades adicionales que se comentaron durante el estudio de viabilidad económica 4.1, ya que permitirían obtener un producto más completo y atractivo para los usuarios.

# Capítulo 6

## Summary and Conclusions

In conclusion, the large amount of data that is available for reuse through Open Data platforms, offers the possibility of creating innovative applications that use processing and visualization as the main tools to generate valuable information. Achieving with it, greater knowledge of the sector where they are applied, in this case tourism, which implies competitive advantages and better decision making.

Throughout this project, a market study has been carried out, with the aim of analyzing the solutions that currently offer tourist data processing. Also, the best technologies and architectures have been selected based on the characteristics of the project. With this, a prototype of a fully functional web application and mobile application has been developed, which meet the expectations that were had at the beginning, providing visualizations of various styles that always use updated data. This has made it possible to create a tool that provides a detailed analysis of four highly relevant aspects for the Canarian tourism sector. In addition, the economic viability of carrying out a project of these characteristics has been demonstrated. Therefore, all the objectives that were established at the beginning of this Final Degree Project have been met.

Finally, as future lines, it would be important to include in the application the additional functionalities that were discussed during the economic viability study 4.1, since they would allow obtaining a more complete and attractive product for users.

# Capítulo 7

## Presupuesto

Aparte del estudio realizado sobre la viabilidad económica del proyecto, también se ha llevado a cabo un presupuesto del coste que ha implicado realizar este Trabajo Fin de Grado. La duración del mismo ha sido aproximadamente de 300 horas, acorde con lo establecido en el plan de estudios de Graduado en Ingeniería Informática, publicado el 21 de marzo de 2011, por la Universidad de La Laguna.

<b>Tareas</b>	<b>Duración</b>	<b>Coste</b>
Análisis del estado actual y tendencia del mercado	25 horas	16 €/h
Estudio y selección de las arquitecturas de la aplicación	40 horas	16 €/h
Desarrollo del Backend	95 horas	20 €/h
Desarrollo del Frontend web y móvil	75 horas	20 €/h
Documentación	65 horas	16 €/h
<b>Total</b>	<b>300 horas</b>	<b>5.480 €</b>

Tabla 7.1: Presupuesto del Trabajo de Fin de Grado

Por tanto, para completar el Trabajo Fin de Grado se han empleado **300 horas**, con un coste total de **5.480 €**.

# Bibliografía

- [1] *Aportación anual del sector turístico al producto interior bruto (PIB) de Canarias entre 2010 y 2020*. Último acceso: 22/08/2022. Statista. [Online]. Available: <https://es.statista.com/estadisticas/526585/aportacion-del-turismo-al-pib-de-canarias/>
- [2] (2021, Mayo) *Estado del arte y tendencias Big Data y Business Analytics*. Último acceso: 22/08/2022. Junta de Andalucía. [Online]. Available: <https://observatorio.andaluciaconectada.es/?wpdmdl=5130>
- [3] *Uso de tecnologías digitales por empresas en España*. Último acceso: 23/08/2022. ONTSI. [Online]. Available: <https://www.ontsi.es/sites/ontsi/files/2022-01/usotecnologiasdigitalesempresas2022.pdf>
- [4] *Previsión de los ingresos de la industria de big data en el mundo entre 2017 y 2027*. Último acceso: 24/08/2022. Statista. [Online]. Available: <https://es.statista.com/estadisticas/517644/prevision-del-valor-de-mercado-del-big-data-en-el-mundo/>
- [5] *Interés a lo largo del tiempo principales frameworks frontend*. Último acceso: 25/08/2022. Google Trends. [Online]. Available: <https://trends.google.com/trends/explore?cat=31&date=today%205-y&q=React%20javascript,Vue%20javascript,Angular%20javascript>
- [6] *Tendencia de los principales frameworks frontend*. Último acceso: 25/08/2022. Stack Overflow Trends. [Online]. Available: <https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Cangularjs>
- [7] J. Shah. (2022, Julio) *Top JavaScript Frameworks to Use in 2022*. Último acceso: 25/08/2022. [Online]. Available: <https://wpwebinfotech.com/blog/javascript-frameworks/>
- [8] D. B. González. (2021, Abril) *Principales tipos de aplicaciones móviles*. Último acceso: 26/08/2022. [Online]. Available: <https://profile.es/blog/tipos-aplicaciones-moviles-ventajas-ejemplos/>
- [9] S. Merenych. (2021, Septiembre) *React Native*. Último acceso: 26/08/2022. [Online]. Available: <https://clockwise.software/blog/react-native-vs-ionic-cross-platform-app-development-frameworks/>
- [10] (2022, Febrero) *El sector turístico de Canarias prevé recuperar en 2022 el 90% de la actividad previa a la covid*. Último acceso: 22/08/2022. Agencia EFE. [Online]. Available: <https://www.efe.com/efe/canarias/economia/el-sector-turistico-de-canarias-preve-recuperar-en-2022-90-la-actividad-previa-a-covid/50001311-4738193>



- [11] (2019, Abril) *How much data is generated each day?* . Último acceso: 22/08/2022. World Economic Forum. [Online]. Available: <https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/>
- [12] *¿Qué es big data?* Último acceso: 23/08/2022. Oracle. [Online]. Available: <https://www.oracle.com/es/big-data/what-is-big-data/>
- [13] *¿Cuántas V debería tener el Big Data?* Último acceso: 23/08/2022. Indra. [Online]. Available: <https://www.indracompany.com/es/blogneo/cuantas-v-deberia-big-data>
- [14] *Worldwide Big Data and Analytics Spending Guide.* Último acceso: 23/08/2022. IDC. [Online]. Available: [https://www.idc.com/getdoc.jsp?containerId=IDC\\_P33195](https://www.idc.com/getdoc.jsp?containerId=IDC_P33195)
- [15] *NH Quality Focus Online.* Último acceso: 24/08/2022. Paradigma Digital. [Online]. Available: <https://www.paradigmadigital.com/proyectos/nh-quality-focus-online/>
- [16] *Smart Steps.* Último acceso: 24/08/2022. Movistar. [Online]. Available: <https://www.movistar.co/web/empresas/big-data/smart-steps>
- [17] *Mabrian.* Último acceso: 24/08/2022. [Online]. Available: <https://mabrian.com/es/>
- [18] *ForwardKeys.* Último acceso: 24/08/2022. [Online]. Available: <https://forwardkeys.com/es/>
- [19] *Dataestur.* Último acceso: 24/08/2022. SEGITTUR. [Online]. Available: <https://www.dataestur.es/>
- [20] (2022, Enero) *Turismo destina 25 millones para digitalizar empresas del sector.* Último acceso: 24/08/2022. [Online]. Available: <https://www.mincotur.gob.es/es-es/GabinetePrensa/NotasPrensa/2022/Paginas/20220110-digitalizacion-empresas-turismo.aspx>
- [21] *datos.gob.es.* Último acceso: 25/08/2022. [Online]. Available: <https://datos.gob.es/es/>
- [22] (2021, Mayo) *Iniciativa Aporta.* Último acceso: 25/08/2022. [Online]. Available: [https://datos.gob.es/sites/default/files/datosgobes/iniciativa\\_aporta\\_-\\_contexto\\_y\\_directrices\\_2021\\_0.pdf](https://datos.gob.es/sites/default/files/datosgobes/iniciativa_aporta_-_contexto_y_directrices_2021_0.pdf)
- [23] *Canarias Datos Abiertos.* Último acceso: 25/08/2022. [Online]. Available: <https://datos.canarias.es/portal/>
- [24] *Highcharts.* Último acceso: 25/08/2022. [Online]. Available: <https://www.highcharts.com/>
- [25] *D3.js.* Último acceso: 25/08/2022. [Online]. Available: <https://d3js.org/>
- [26] *Chart.js.* Último acceso: 25/08/2022. [Online]. Available: <https://www.chartjs.org/>
- [27] *React.* Último acceso: 25/08/2022. [Online]. Available: <https://es.reactjs.org/>
- [28] *Angular.* Último acceso: 25/08/2022. [Online]. Available: <https://angular.io/>
- [29] *Vue.js.* Último acceso: 25/08/2022. [Online]. Available: <https://vuejs.org/>
- [30] *Node.js.* Último acceso: 26/08/2022. [Online]. Available: <https://nodejs.org/es/>

- [31] *Express*. Último acceso: 26/08/2022. [Online]. Available: <https://expressjs.com/es/>
- [32] *MongoDB*. Último acceso: 26/08/2022. [Online]. Available: <https://www.mongodb.com/es>
- [33] *React-Bootstrap*. Último acceso: 26/08/2022. [Online]. Available: <https://react-bootstrap.github.io/>
- [34] *Axios*. Último acceso: 26/08/2022. [Online]. Available: <https://axios-http.com/es/docs/intro>
- [35] *node-cron*. Último acceso: 26/08/2022. [Online]. Available: <https://github.com/kelektiv/node-cron>
- [36] *mongoose*. Último acceso: 26/08/2022. [Online]. Available: <https://mongoosejs.com/>
- [37] *Ionic*. Último acceso: 26/08/2022. [Online]. Available: <https://ionicframework.com/>
- [38] *React Native*. Último acceso: 26/08/2022. [Online]. Available: <https://reactnative.dev/>
- [39] *Android Studio*. Último acceso: 26/08/2022. [Online]. Available: <https://developer.android.com/studio>
- [40] *Heroku*. Último acceso: 26/08/2022. [Online]. Available: <https://www.heroku.com/>
- [41] *Digital Ocean*. Último acceso: 26/08/2022. [Online]. Available: <https://www.digitalocean.com/>
- [42] *Namecheap*. Último acceso: 27/08/2022. [Online]. Available: <https://www.namecheap.com/>
- [43] *API REST*. Último acceso: 28/08/2022. IBM. [Online]. Available: <https://www.ibm.com/es-es/cloud/learn/rest-apis>
- [44] *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*. Último acceso: 28/08/2022. BBVA. [Online]. Available: <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/>
- [45] *express.Router*. Último acceso: 30/08/2022. Express. [Online]. Available: <https://expressjs.com/en/guide/routing.html#express-router>
- [46] *Códigos de estado de respuesta HTTP*. Último acceso: 30/08/2022. Mozilla Web docs. [Online]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Status>
- [47] *morgan*. Último acceso: 30/08/2022. [Online]. Available: <https://github.com/expressjs/morgan>
- [48] *cors*. Último acceso: 30/08/2022. [Online]. Available: <https://github.com/expressjs/cors>
- [49] *react-router*. Último acceso: 31/08/2022. [Online]. Available: <https://github.com/remix-run/react-router>
- [50] *pm2*. Último acceso: 01/09/2022. [Online]. Available: <https://pm2.keymetrics.io/>

- [51] *NGINX*. Último acceso: 02/09/2022. [Online]. Available: <https://www.nginx.com/>
- [52] *Let's Encrypt*. Último acceso: 02/09/2022. [Online]. Available: <https://letsencrypt.org/es/>
- [53] (2019, Septiembre) *Choosing the Right Droplet Plan*. Último acceso: 03/09/2022. [Online]. Available: <https://docs.digitalocean.com/products/droplets/concepts/choosing-a-plan/>
- [54] *DigitalOcean Pricing*. Último acceso: 03/09/2022. [Online]. Available: <https://www.digitalocean.com/pricing/droplets>