Trabajo de Fin de Grado

Análisis de los incidentes del CECOES 1-1-2 utilizando técnicas de Ciencia de los Datos

Analysis of emergency incidents using Data Science techniques

Teno González Dos Santos

La Laguna, 6 de junio de 2016

D. **Marcos Colebrook Santamaría**, con N.I.F. 43.787.808-V, Profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

D. **Carlos J. Pérez González**, con N.I.F. 45.452.719-G, Profesor Asociado de Universidad adscrito al Departamento de Matemáticas, Estadística e Investigación Operativa de la Universidad de La Laguna, como cotutor.

C E R T I F I C A N

Que la presente memoria titulada:

"Análisis de los incidentes del CECOES 1-1-2 utilizando técnicas de Ciencia de los Datos"

ha sido realizada bajo su dirección por D. **Teno González Dos Santos**, con N.I.F. 54.059.118-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de marzo de 2016.

Agradecimientos

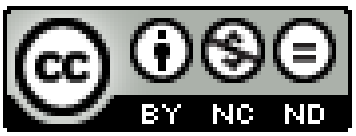
En especial a mi tutor en este proyecto, Marcos Colebrook Santamaría por su gran apoyo y especial interés en el proyecto. Para este proyecto fueron claves su gran experiencia, su paciencia infinita, sus ganas de enseñar e incluso, de aprender, y por encima de todo, su calidad humana.

Especial agradecimiento para Carlos J. Pérez González, cotutor del proyecto por contagioso interés en aprender y buscar soluciones, su tranquilidad y disposición a ayudar, que tan necesarios fueron en muchos momentos del proyecto.

También agradecer a José Luis Roda García, colaborador del proyecto por su entusiasmo, ganas y confianza en la capacidad de los alumnos. Una energía que contagia a los de su alrededor.

Destacar la inestimable ayuda de Carlos B. Rosa Remedios, Responsable de la Unidad de Tecnologías de la Información y la Comunicación en el CECOES 1-1-2, por la confianza depositada. No es fácil "pelearse" con un organismo público para sacar proyectos de este tipo adelante, pero siempre puso todos los medios a su alcance para lograrlo.

Licencia



Aviso legal

Resumen

El objetivo de este trabajo ha sido el establecer un acercamiento gradual y con una dificultad progresiva a los campos de la Ciencia de los Datos y el *Big Data* a través de un enfoque eminentemente práctico.

La principal razón que llevó a la elección de este tema para la elaboración del proyecto es la relevancia manifiesta que está tomando en la sociedad actual el eficiente manejo y tratamiento de unos datos cada vez más masivos y heterogéneos, con el fin de obtener información útil para las organizaciones.

Para ello, desde un comienzo se estableció la posibilidad de desarrollar el proyecto en colaboración con una empresa externa al ámbito puramente académico, con el fin de dotar de valor y utilidad al conjunto del trabajo de investigación y desarrollo llevado a cabo. Estas empresas serían las encargados de nutrirnos con datos reales y problemáticas a resolver igualmente verídicas, a la vez que nos establecerían una serie de restricciones y condiciones a la hora de trabajar.

A través de los tutores del proyecto, Marcos Colebrook y Carlos J. Pérez, y del colaborador José L. Roda, surgió la posibilidad de colaborar con el Centro Coordinador de Emergencias y Seguridad (CECOES) 1-1-2 del Gobierno de Canarias. De esta manera, y gracias a la ayuda de Carlos B. Rosa Remedios, Responsable Unidad de Tecnologías de la Información y la Comunicación en el CECOES 1-1-2 de Canarias, pudimos establecer un marco claro de trabajo con unos requisitos y restricciones definidos, unos datos de entrada verídicos y unos objetivos prácticos, a la vez que flexibles, en las diferentes fases que se irían completando de manera progresiva, como se verá más adelante.

Teniendo claro el objetivo, y con datos sobre los que trabajar, se abordó la problemática de la elección del software como solución, elementos tangenciales como puede ser el lenguaje y entorno de desarrollo, la metodología a seguir, los requisitos de hardware necesarios y un largo etcétera.

Por último, se procedió a la realización del proyecto siguiendo una metodología iterativa e incremental, acompañada por diversas reuniones que sirvieron para darle forma a la aplicación final.

Palabras clave: Ciencia de los Datos, Big Data, R, CECOES, 1-1-2, Shiny, RStudio.

Abstract

The aim of this research project has been to establish a gradual introduction to both the field of Data Science and Big Data, with a progressive difficulty, using a completely practical approach.

The main reason for the choice of the topic of this project is the obvious relevance in today's society of an efficient manipulation and processing of data which is continuously growing and becoming more varied, with the aim of obtaining useful information for organizations.

In order to do that, since the beginning there was the possibility of carrying out the project in collaboration with a company, external to the academic context, for the purpose of giving added value and usefulness to the whole research and development project.

These companies were in charge of feeding us both real data and real problems to solve, and at the same time, they established a series of restrictions and conditions to work with.

The project tutors, Marcos Colebrook and Carlos J. Pérez, came out with the possibility of working with the Emergency Service of the Canary Islands (CECOES 1-1-2) and the ICT services at the University of La Laguna.

Thus, thanks to the help of Carlos B. Rosa Remedios, head of the IT and Telecommunications Unit at the CECOES 1-1-2, we were able to establish a clear framework with some defined preconditions and restrictions, some real input data and some practical (also flexible) goals in the stages of the project we were gradually carrying out, as we will be see below.

Having a clear goal and data to work with, we addressed the problem of choosing of a software solution, a programming language and a proper development environment, a methodology to follow, the hardware requirements and a large etcetera.

Finally, we proceed to the project implementation following an iterative and incremental methodology, along with several meetings that helped us to shape our project application.

Keywords: *Data Science, Big Data, R, CECOES, 1-1-2, Shiny, RStudio.*

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Justificación del proyecto

1.1 Introducción

El tema del proyecto, Ciencia de los Datos (*Data Science*), hace mención a un campo interdisciplinario que involucra los procesos y sistemas para extraer conocimiento o un mejor entendimiento de grandes volúmenes de datos en sus diferentes formas (estructurados o no estructurados), y sus respectivos formatos.

Este proyecto, por lo tanto, se justifica en base a la situación tecnológica actual por la cual se generan datos a mayor velocidad de la que se puede procesar. Ante este problema, las grandes empresas no ven más alternativa que eliminarlos ante la imposibilidad de su almacenamiento. Sin embargo, gracias al fenómeno del *Big Data* (datos masivos) y del *Data Science*, ya es posible extraer información útil tras procesar estos datos.

Big Data y *Data Science* son campos relativamente nuevos, y que en España su uso se encuentra aún en una fase inicial y poco extendida, si observamos el gráfico extraído del estudio realizado por *Wikibon* [14] -una comunidad internacional de profesionales de la tecnología- (ver Figura 1.1) en cual trata de predecir el volumen de mercado generado por negocios o proyectos *Big Data* hasta 2026. Podemos observar la clara relevancia de este fenómeno dentro del contexto mundial actual.

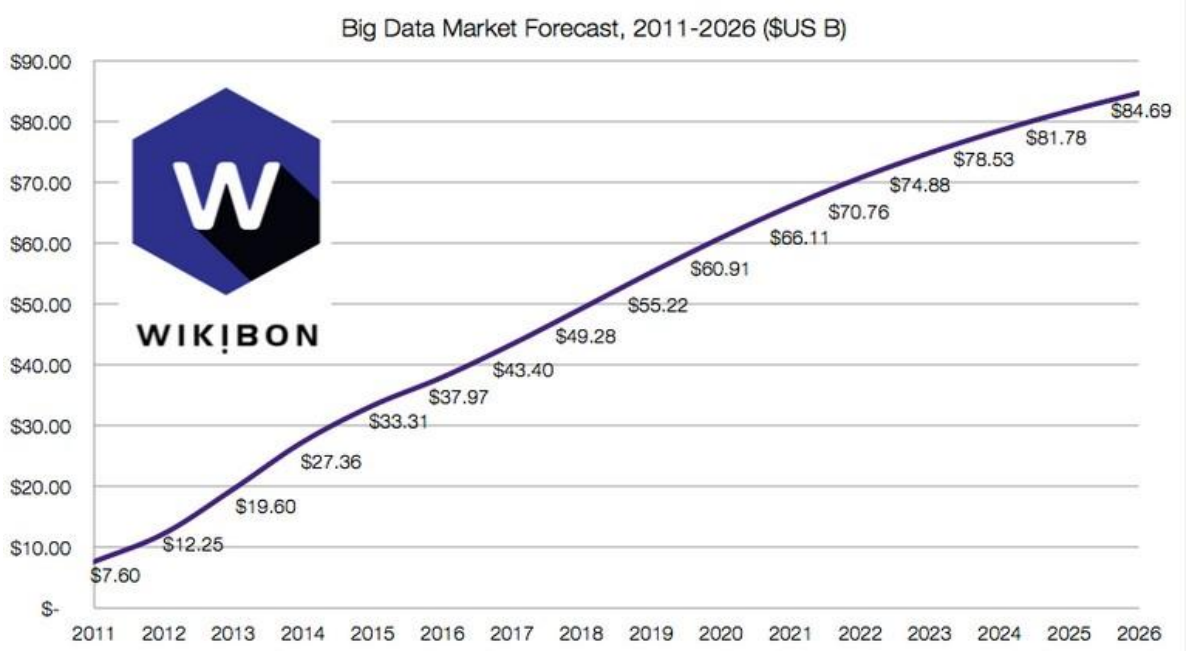


Figura 1.1. Gráfica de predicción de crecimiento de mercado *Big Data* (billones de dólares).

1.2 Desarrollo colaborativo con CECOES 1-1-2 Canarias

Durante las reuniones iniciales para el planteamiento del proyecto, surgió la posibilidad de trabajar de manera colaborativa con el Centro de Coordinador de Emergencias y Seguridad (CECOES) 1-1-2 del Gobierno de Canarias, y por ende, realizar un proyecto con datos reales (anonimizados), problemáticas reales y sobre todo, con resultados que pudieran tener utilidad real.

El objetivo principal que se estableció en las diferentes reuniones llevadas a cabo con personal del 1-1-2 fue el de crear una interfaz que pudiera mostrar información extraída de los datos, ya fuera en forma de gráficas dinámicas o mapas. De esta manera se podrían detectar patrones de incidentes, tendencias o anomalías en el funcionamiento de los servicios de emergencia en Canarias. Siendo todo ello posible a partir de la utilización de herramientas de *Data Science*.

Además, el proyecto serviría como un puente de cara a futuras colaboraciones entre el Grado de Ingeniería Informática y el CECOES, ya fuera en proyectos derivados del actual o completamente nuevos.

Capítulo 2.
Estado del arte

2.1 Introducción

Con el fin de establecer un marco teórico claro sobre el que trabajar definiremos qué es ***Data Science*** y ***Big Data***, su influencia y relevancia actual, así como una serie de características propias del caso de estudio que nos atañe y la resolución del mismo.

A diario, en el mundo se generan 2.5 trillones de bytes de información, tanto es así que el 90% de los datos a nivel mundial se han creado solamente en los últimos 2 años. Esta información proviene de todos lados, sensores que recogen información climática, publicaciones en las redes sociales, imágenes y vídeos digitales, registros de compra y transacciones y señales de GPS de los móviles, entre otros. Toda esta información se conoce como *Big Data*, y es a partir de esta fuente masiva de datos la razón por la cual es inminente el nacimiento de un profesional que conozca y genere un uso a esta información: el **Científico de Datos**.

El Científico de Datos es una nueva profesión que hoy es considerada clave en el mundo de las tecnologías, además de ser una de las mejor pagadas. Se trata de una persona formada en las ciencias matemáticas y estadísticas, que domina la programación y sus diferentes lenguajes, así como las ciencias de la computación y la analítica.

El profesional de Ciencia de los Datos también debe tener la capacidad y los conocimientos necesarios para comunicar sus hallazgos a medida que los tiene, no sólo al área de tecnología sino además al sector de los negocios. Debe dominar la tecnología y las bases de datos para modificar y mejorar la orientación de los negocios de la empresa para la que trabaja.

El Científico de Datos analiza, interpreta y comunica las nuevas tendencias en el área y las traduce a la empresa para que ésta haga uso de ellas y adapte sus productos y servicios, y cree nuevas oportunidades de negocio. Google, por ejemplo, tiene 600 personas dedicadas al estudio del *Big Data*.

Sin embargo, la selección de un equipo de *Big Data* no es tarea sencilla: un reciente estudio de la **Unidad de Inteligencia de la revista *The Economist*** [15] entrevistó a 600 ejecutivos globales y el 54% de los empresarios estadounidenses aseguraron que encontrar a los profesionales adecuados para un exitoso proyecto de *Big Data* es el obstáculo más importante para no hacerlo.

2.2 Software Estadístico *R*

Desde el punto de vista del *framework* utilizado para el desarrollo del proyecto, éramos conscientes de la necesidad de una herramienta que aunara ambos perfiles del proyecto, estadístico e informático. Además, existía la obvia necesidad de que pudiera cargar grandes cantidades de datos con rapidez, además de manipularlos y procesarlos a igual velocidad.

La primera gran decisión en este sentido llegó a la hora de elegir un software que cumpliera todo esos requisitos. A pesar de la existencia de diferentes software estadísticos, tanto de pago (*SPSS*, *Mathematica* de *Wolphram,* etc) como gratuitos (librerías de *Python*, *MatLab, etc*), el conjunto de los miembros del proyecto (tutores y alumno) tuvo claro la decisión final de elegir como *framework* el software estadístico *R*.

2.2.1 ¿Qué es *R*?

R es lenguaje y entorno de programación que permite análisis estadístico y gráfico. Se trata de un proyecto de software libre de GNU similar al lenguaje y entorno *S*, desarrollado en los laboratorios Bell por John Chambers y otros investigadores. Fue inicialmente desarrollado por Robert Gentleman y Ross Ihkaka del Departamento de Estadística de la Universidad de Auckland en 1993, uniendo la principales fortalezas de los lenguajes *S* y *Scheme*.

Es un lenguaje ampliamente utilizado a nivel mundial en investigación biomédica, bioinformática y matemáticas financieras, entre muchas otras áreas de trabajo.

R posee una gran variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series temporales, etc) y gráficas, y es, además, altamente extensible.

Una de los puntos fuertes de *R* es la facilidad con la que produce gráficos bien diseñados y de calidad. Además, existe una gran capacidad de personalización de las gráficas y los elementos interactivos, dotando al usuario de gran control.

2.2.2 El entorno *R*

R es, además, un conjunto de herramientas software para la manipulación, procesamiento y representación gráfica de datos. Incluye:

- capacidad para gestionar y almacenar datos de forma efectiva
- un conjunto de operadores para cálculos sobre vectores y matrices
- una gran colección de herramientas para el análisis de datos
- herramientas gráficas para la representación y el análisis de datos
- un lenguaje bien desarrollado con toda la potencia necesaria: bucles, funciones recursivas, condicionales, gestión de entrada y salida, etc.

El término "entorno" trata de establecerlo como un sistema completamente planificado y coherente, más que como una recopilación incremental de herramientas específicas e inflexibles, como es el caso con otros software de análisis de datos.

Además, para tareas de computacionalmente intensas, se usa código *C, C++* o *Fortran* durante la ejecución para acelerar las mismas. De la misma forma, los usuarios avanzados pueden escribir sus propias rutinas en *C* para manipular objetos *R* directamente.

2.2.3 ¿Por qué *R*?

Existen diferentes factores que nos llevaron a elegir *R* como el entorno y lenguaje adecuado para nuestro proyecto. Entre los aspectos determinantes en la decisión se encuentran los siguientes:

Software libre

La principal razón que nos llevo a decidirnos por este software estadístico es el hecho de tratarse de un proyecto de software libre y que, por lo tanto, evitaba cualquier problema de licencias de producto, distribución, etc, que podrían surgir usando otros programas.

Comunidad de desarrolladores

El hecho de ser software libre conlleva sus propias ventajas, como la existencia de una gran comunidad internacional implicada en el desarrollo y soporte del lenguaje. Sin duda, esta virtud fue determinante la elección, pues en el proyecto se han usado librerías y módulos que no se encontraban en el software básico y que han sido desarrollado por usuarios de la comunidad.

Facilidad de uso

En este aspecto, *R* se adaptaba perfectamente a nuestras necesidades al tratarse de un lenguaje orientado a objetos, lo cual hacía mucho más sencillo adecuarse a su programación al estar acostumbrado a este tipo de lenguajes durante el resto de la carrera.

Además, en este sentido *R* cuenta con un entorno gráfico llamado *RStudio*, que permite realizar cualquier operación rutinaria a través de menús y ventanas, facilitando mucho la labor del desarrollador.

2.3 *R + Shiny*

Una vez se estableció el lenguaje y entorno de programación que se iba a usar para la carga y procesamiento de los datos era necesario decidir elegir una herramienta que nos permitiera crear una aplicación gráfica en la cual pudiéramos representar los datos agrupados y mostrar los resultados de forma gráfica.

Esta elección resulto fácil y rápida, pues ya teníamos conocimiento de un *framework* de aplicaciones web interactivas que se apoya en *R* para ejecutar las mismas.

2.3.1 ¿Qué es *Shiny*?

Shiny es una herramienta para crear fácilmente aplicaciones web interactivas que permiten a los usuarios interactuar con sus datos sin tener que manipular el código. La programación Reactiva en la que se basa enfatiza el uso de:

- Valores que cambian en el tiempo.
- Expresiones que registran esos cambios.

De esta forma, la aplicación se adapta a las selecciones del usuario y cambia la representación de los datos en la interfaz de manera interactiva.

2.3.2 ¿Por qué *Shiny*?

La elección de este paquete de *RStudio* como herramienta de creación de aplicaciones interactivas se basa en varios puntos fundamentales:

Integración con *R*

Ya que se trata de un paquete de *RStudio*, se puede escribir el código de la aplicación directamente en lenguaje *R*, lo que nos permite acceder y manipular nuestros datos directamente.

Versatilidad

Una de la grandes virtudes de *Shiny* es que el código de la interfaz no solo admite programación en lenguaje *R*, sino que se puede desarrollar la misma usando lenguajes como *HTML*, *JavaScript* o *CSS* para mayor flexibilidad.

Desarrollo comunitario

El hecho de ser un paquete de *R*, permite a esta herramienta beneficiarse de todas sus ventajas, como es el caso del desarrollo por parte de la comunidad. Esto a permite usar increíbles gráficos y mapas interactivos, los cuales, ajustándolos para satisfacer las necesidades de nuestro proyecto, han dado un salto de calidad enorme a la visualización de los datos.

Capítulo 3. Problemática

A la hora de enfocar el proyecto se optó por darle una vertiente eminentemente práctica y realizarlo en colaboración con empresas o entidades, las cuales serían las encargadas de proveer de datos sobre los que trabajar así como establecer las tareas a desarrollar.

En este contexto surgió la posibilidad de trabajar en colaboración con el **Centro Coordinador de Emergencias y Seguridad (CECOES 1-1-2) del Gobierno de Canarias**, el cual es un servicio público que nació el 30 de mayo de 1998 para dar respuesta a todas las llamadas de emergencia que se producen en el archipiélago canario.

El CECOES requería una aplicación analizara la gran cantidad de registros de llamadas que se reciben día tras día en los centros canarios. De esta forma, existían almacenados datos de los 18 años de existencia del centro de emergencias en las islas.

Para registrar todas las llamadas de incidencias producidas el centro contaba con una aplicación de escritorio basada en un formulario. Según se desarrolla una llamada, un operador del centro va registrando la información del incidente rellenando diferentes campos, que quedan almacenados como un único registro en una base de datos global.

Este sistema ya llevaba bastante tiempo en uso y, como comentó el propio representante de CECOES durante las reuniones iniciales, se estaba implantando una nueva herramienta más moderna durante el desarrollo de este propio proyecto.

De esta manera se advierte de la clara necesidad de sacar información útil de todos esos datos de registros de llamadas almacenados durante años. Por lo tanto, se plantea la posibilidad de llevar a cabo un proyecto basado en *Big Data* y análisis estadístico, mediante el cual procesen, analicen y representen los datos almacenados por el centro durante los últimos 10 años y se cree una

herramienta que permite visualizar los resultados a través de una interfaz interactiva.

3.1 Punto de partida y alcance

A raíz de una serie de reuniones con el responsable de la Unidad de Tecnologías de la Información y la Comunicación en el 1-1-2 se planteó la problemática total sobre la que trabajaríamos y se establecieron una serie de hitos y objetivos a cumplir que, en gran medida, crearon la estructura de este proyecto.

Estos encuentros incluyeron una visita al propio Centro de Emergencias, así como una reunión con el personal que gestiona los datos del CECOES, que daría una visión global de los datos que poseían, así de su obtención y almacenamiento.

Durante esta visita al centro se pudo ver cómo era el proceso que se seguía para registrar cada llamada (incluso se pudo presenciar la gestión de un incidente en directo). Se trató de una visita muy productiva, pues permitió conocer de primera mano la fuente de los datos y la organización que se utiliza, así como los protocolos que se siguen para cumplimentar la información que reciben de cada llamada.

Desde el CECOES se dejó mucha libertad a la hora de establecer los **objetivos** y el **alcance** del proyecto. Ellos, como proveedores de datos, establecieron una guía de lo que esperaban conseguir; esto es, una interfaz gráfica interactiva que les permitiera filtrar datos incluso por municipio, tipo de incidente, año, mes, etc., y que, por lo tanto, pudieran distinguir patrones o tendencias en los incidentes a lo largo de tiempo. De esta forma se concreta unas líneas maestras para el proyecto, pero a su vez dejando mucha libertad para estructurar el proyecto, manipular los datos y crear la interfaz.

Además, durante varias reuniones con personal de CECOES se pudo observar el estado de los datos y establecer así un **punto de partida** para el proyecto. Se pudo hacer un esquema de todos los campos almacenados durante el registro de las llamadas, así como del tipo de estructura de datos. De la misma forma se nos comunicó el proceso de almacenamiento de los datos, los cuales estaban almacenados en una base de datos administrada con

Microsoft Access, lo cual podría llevar consigo alguna dificultad para extraer los datos y llevarlos a nuestra herramienta elegida, *R*.

3.2 Metodología de trabajo

Uno de los mayores retos cuando se trabaja con grandes cantidades de datos es verificar la exactitud de los resultados obtenidos, siendo éste un aspecto crítico para el personal del CECOES.

De esta manera se diseñó una metodología de trabajo por la cual se llevaría a cabo un desarrollo dividido en diferentes fases basadas en las distintas operaciones desde que se reciben los datos "en crudo" desde el 1-1-2, hasta que se termina el último detalle de la interfaz interactiva. De esta forma se adoptó por acotar a 10 años la cantidad de datos suministrados, es decir, **más de 7 millones de registros** (ver Tabla 3.1).

Estado inicial de los datos	
Número de registros total	7.003.168
Número de variables original	21
Número de datos original (filas*columnas)	147.066.528
Número de variables creadas tras limpieza y procesamiento	7
Número de datos final tras limpieza y procesamiento (filas*columnas)	196.088.704

Tabla 3.1. Información sobre el estado inicial de los datos

Consecuentemente, el desarrollo se fragmentó en hitos, cada uno de los cuales, como se ha establecido anteriormente, atendía a cada una de las fases de manipulación de los datos, así como de programación de la interfaz gráfica. Así, se da evidencia de la libertad dada por el CECOES para estructurar el proyecto, pues tras recibir los datos originales (anonimizados) se permitió una organización propia y libre (aunque controlada) con posteriores reuniones que servirían para evaluar el estado del proyecto durante las diferentes fases.

Haciendo un símil y salvando las distancias, la metodología de "Desarollo Interativo e Incremental", siendo esta un conjunto de tareas agrupadas en etapas repetitivas (iteraciones) que permiten al desarrollador ir "incrementando" el producto aprovechando los conocimientos aprendidos en el desarrollo anterior, creando un producto final cada vez más completo.

La plataforma usada para la gestión del proyecto por parte de los miembros del proyecto (profesores y alumno) fue Google Drive, permitiendo compartir documentación o cualquier elemento de interés que pudiera aportar valor o solucionar cualquier dificultad surgida durante la elaboración del proyecto.

Para llevar a cabo el proyecto se definieron las siguientes **fases**:

- **Exportación los datos** del sistema de base de datos original bajo *Microsoft Access* y conseguir los mismos archivos en formato "csv" (*comma-separated values*). El resultado final son 20 ficheros (por año y provincia).
- Realizar una **limpieza de los datos** para dotarlos de la máxima homogeneidad, cohesión y coherencia posibles. Esto incluye un proceso de **análisis inicial** del estado original de los datos, sus campos y valores, así como de la necesidad de **normalizar** los mismos, es decir, que los valores para cada campo vayan estipulados por una serie de parámetros marcados que permitan evitar inconsistencias. Esta etapa fue la más duradera y exhaustiva del proyecto debido la heterogeneidad encontrada en los datos.
- **Análisis de los datos** recogidos en los últimos 10 años. Durante esta etapa se observan los campos y estructuras de datos utilizados con el fin de decidir qué campos se consideran más relevantes o de cuáles se puede extraer mayor cantidad de información útil y que formen parte de los resultados a mostrar en la interfaz gráfica interactiva.
- **Creación de un archivo único** que una todos los anteriores, ya homogeneizado y estructurado de la forma deseada. Este archivo se considerará el punto de partida para crear subconjuntos que se utilicen para actividades más específicas (análisis de determinados tipos de incidentes, años o períodos concretos, etc.).
- **Diseño y análisis de todas funcionalidades que va a ofrecer la interfaz** de usuario, basándonos en la naturaleza de los datos, así como

en los objetivos marcados por el propio 1-1-2. Esto incluye establecer todas las interacciones que podrá realizar el usuario, así como concretar las diferentes respuestas a cada selección que realice el usuario.

- **Implementación de la interfaz gráfica interactiva.** Esta etapa desarrollada (como todas, pero ésta en especial) de forma iterativa e incremental, incluiría la creación del mapa del archipiélago que se podrá filtrar según selecciones del usuario, así como gráficos interactivos y una nube de términos.

Cada una de estas fases, como ya se ha establecido, se irían repitiendo varias veces, ya sea por contrastar la correcta carga de los datos, analizar los datos en busca de nuevas posibilidades para la interfaz o directamente seguir aumentando la funcionalidad de ésta, y siempre de forma iterativa e incremental.

Capítulo 4.
Fases y desarrollo del proyecto

Durante este capítulo se describirán todas las etapas del proyecto de forma pormenorizada, desde que se instala el software que dará soporte al proyecto, hasta el último detalle en el acabado de la interfaz gráfica de usuario.

4.1 Instalación de *R*, *RStudio*, librerías necesarias y *Shiny*

El primer paso en todo desarrollo es la instalación de las dependencias y requerimientos para llevar a cabo el proyecto.

Lo primero de todo se instaló la herramienta que daría toda la fuerza estadística y de programación al proyecto: el lenguaje y entorno *R*. Al tratarse de un proyecto de software libre es fácilmente descargable desde su página oficial. Simplemente hay que seleccionar un *mirror* (lo más cercano a tu ubicación, a ser posible), elegir la plataforma de desarrollo (en mi caso *Windows 8.1* 64bits) y el paquete base (pues no vamos a desarrollar ninguna librería para la comunidad).

Una vez instalado *R*, es necesario un Entorno de Desarrollo Integrado (*IDE*) que facilite la navegación entre directorios, la manipulación y visualización de los datos, etc. Resulta muy importante, pues R no posee más que una interfaz por comando, dificultando proyectos de mayor complejidad. Por lo tanto, es preciso incluir en nuestro sistema *RStudio*. Este *IDE* es fácilmente descargable desde su página oficial, siguiendo las mismas directrices que en *R* para nuestro sistema.

Tras tener ya nuestro entorno instalado, es preciso descargar las librerías que se necesitarán para manipulación de datos a gran escala. Para ellos

usaremos el instalador de paquetes que provee el entorno a través de un menú contextual bastante sencillo. Algunos ejemplos de estos paquetes de librerías son *dplyr*, que posee diferentes funciones de agregación y filtrado de datos; o *highchartsUtils* y *rCharts*, para la representación del mapa y los gráficos. En la siguiente imagen se ve un ejemplo del uso de la librería *dplyr*:

```
table.dia_by_muni_by_sexo<-112_ffdf %>%
     filter(anio="2009") %>%
     group_by(dia,sexo,municipio) %>%
     summarize(n=n()) %>%
     mutate(prop.catch=n/sum(n)) %>%
     arrange(desc(prop.catch))


##     row          MES   SEXO MUNICIPIO TOTAL
##       1     01-ENERO HOMBRE     ADEJE     1
##       2     01-ENERO  MUJER     ADEJE     1
##       3     01-ENERO HOMBRE    AGAETE     1
##       4     01-ENERO  MUJER    AGAETE     1
##                  ...    ...       ...   NaN
##   69349 31-DICIEMBRE HOMBRE     YAIZA     1
##   69350 31-DICIEMBRE  MUJER     YAIZA     1
```

Figura 4.1. Ejemplo de filtrado y agregación de datos usando *dplyr*.

De la misma forma se instala *Shiny*, el paquete que permite la creación de una aplicación web gráfica e interactiva. A través del gestor de paquetes es fácilmente descargable y listo para su uso.

4.2 Recepción y estado original de los datos

Esta etapa trata de describir el proceso que se siguió tras recibir los datos tal cual se almacenaban en el sistema de base de datos original y exportarlos a nuestro entorno.

Los archivos originalmente estaban almacenados en un sistema de base de datos con *Microsoft Access*. Como es evidente, se trata de una versión muy antigua y, por lo tanto, con versiones más recientes, existían inconsistencias

en campos como la fecha debido a diferencias en los tipos de estructuras de datos.

Para solucionar este problema se optó por extraer directamente los datos en ficheros *.csv*, es decir, cada campo separado por comas (punto y coma en nuestro caso, por los decimales). Este proceso dio como resultado 20 ficheros (por año y provincia) de 200 Mb. Cada uno de estos ficheros tardaba unos 15 segundos en cargar en el entorno.

En el siguiente gráfico se puede comprobar el cálculo de los tiempos de carga de los ficheros originales en *.csv:*

```
start <- proc.time()
datos.LPGC.2005 <- read.csv("datosSCTF2005.csv", sep=";",
dec=",", stringsAsFactors=FALSE)
end <-proc.time() - start

## [1] "Fichero incidencias 2005: 14.43 segundos"
## [1] "Fichero incidencias 2006: 18.9 segundos"
## [1] "........."
## [1] "Fichero incidencias 2014: 15.9 segundos"
```

Figura 4.2. Medición de tiempos de carga de ficheros *.csv* originales.

La anterior función (entre medidores de tiempo) carga un fichero de formato *csv* usando como separador el punto y coma, y dejando las cadenas de caracteres como tal (no factores). El resultado de esta carga es asignado a una variable.

Estado original de los datos: Análisis inicial

Previamente a la limpieza y depuración de los datos era preciso realizar un análisis inicial del estado original de los datos. Este análisis pretendía entender cada uno de los campos y lo que representaba. Además, se reflexionó sobre la utilidad o no de ciertos campos, así como la posibilidad de añadir algunos nuevos, surgidos de combinación de los mismos o de alguna fuente externa.

A continuación se enumeran y describen los diferentes campos que existían en los datos originales:

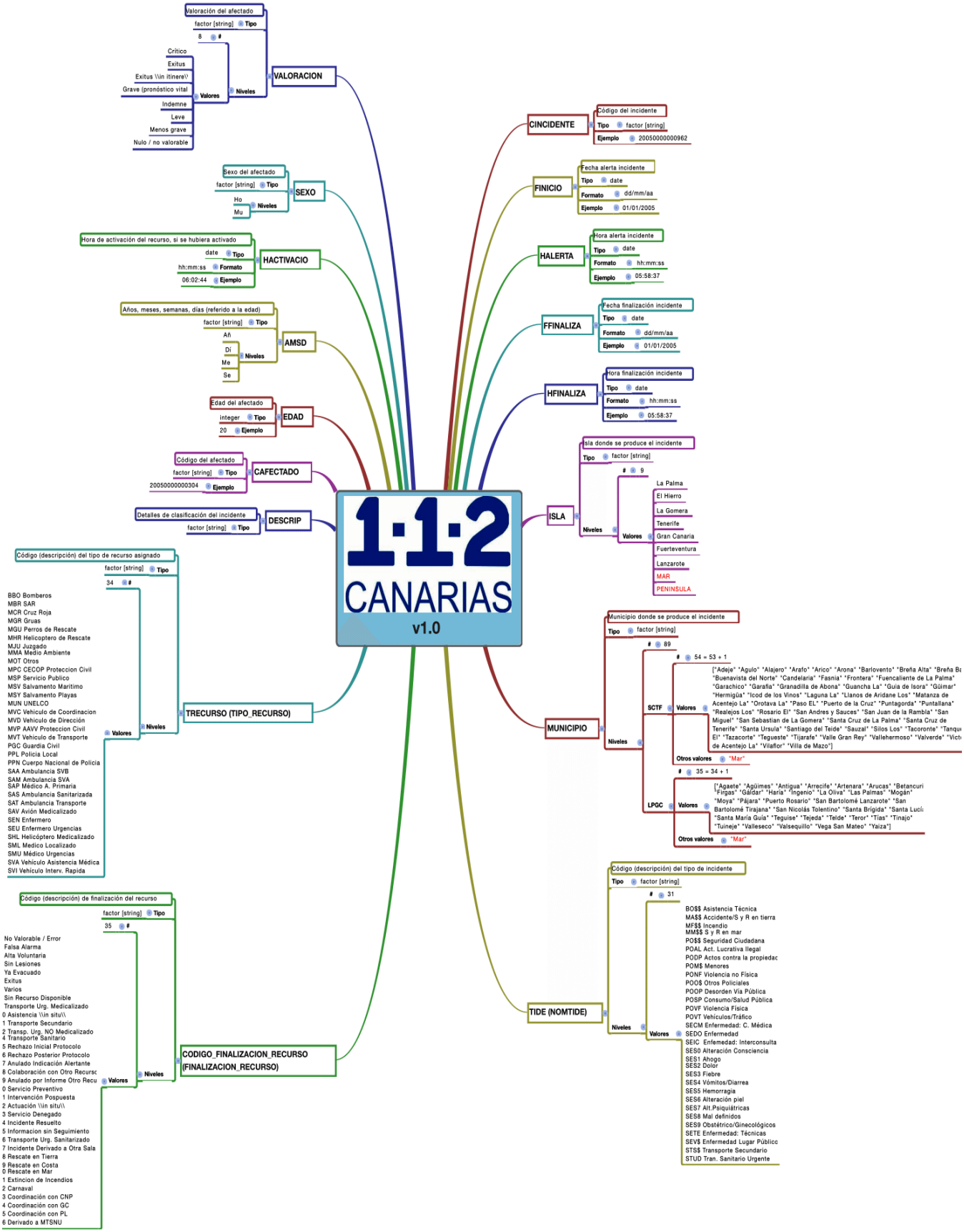


Figura 4.3. Desglose de campos.

4.3 Limpieza de datos

Este proceso se basa en dotar a los datos de la máxima homogeneidad, cohesión y coherencia posibles. Esto incluye un proceso de **normalización** de los mismos, es decir, que los valores para cada campo vayan estipulados por una serie de parámetros marcados que permitan evitar inconsistencias.

Tras hacer algunas pruebas con comandos como *summary* o *describe*, se pudieron ver inconsistencias en valores de ciertos campos (principalmente los de tipo cualitativo). A lo largo de los registros de varios campos, se daban valores que no reflejaban ninguna realidad, ni aportaban ningún valor al campo. Este fenómeno se principalmente a tres factores:

- **Fallo humano**: como no podía ser de otra forma, cuando se precisa máxima rapidez y efectividad a la hora de registrar cada incidente desde el centro del 1-1-2 es lógico que alguna vez que se introduzcan datos de forma errónea o con incorrecciones. Aun así, se debe dejar claro que estos son gran minoría, pues el trabajo de los operadores es excelente, aunque se deben tener en cuenta para aportar consistencia a los datos y, sobre todo, que no se propague el error hacia el resultado final.
- **Conversiones de datos entre sistemas**: es posible que cuando la herramienta de registro procesa a información del formulario y realiza la petición de inserción en la base de datos, haya existido algún problema por divergencias en el tipo de datos, llevando al sistema a realizar una conversión que podría dar lugar a inconsistencias en el dato final.
- **Cambios en la herramienta de registro**: debido a lo longevo de los datos que estamos manipulando, estos han pasado por diferentes etapas y reestructuraciones en la forma en la que se recogen los datos. Estos cambios han podido llevar a que se registran campos iguales de diferentes formas o con diferentes denominaciones. A pesar de que de nuevo estas inconsistencias no representaban una cantidad tan significativa como para influenciar en el resultado final, si que era preciso normalizar estos campos para dotar, de nuevo, de mayor homogeneidad y consistencia a los datos.

Pasos a seguir

Para este proceso se consideró, como parece lógico, unir todos los archivos en un único fichero que aúne la totalidad de los datos. Sin embargo, habían varias restricciones que impedían realizar este paso:

- la necesidad de saber de qué provincia era cada archivo, pues se iba a crear un campo con el código de la misma. Por lo tanto, para esto era necesario extraer la información del propio nombre del fichero.
- resultaba útil tener los ficheros depurados y normalizados de forma individual por año y provincia, en caso de precisarlos posteriormente.

A continuación se enumeran los pasos seguidos para la depuración y normalización de cada uno de estos ficheros (en el apéndice se encuentra el código completo de limpieza):

Normalización de los nombres de las variables

En este primer paso, se pretende dar un formato más amigable a las denominaciones de cada campo. Inicialmente, algunos campos están en mayúsculas, otros en minúsculas, otros con espacios en blanco, etc. Esto no resulta muy cómodo para manipular posteriormente nuestros datos. Por lo tanto, se sustituyen espacios en blanco por "_" y ponen todas las letras en mayúscula, dotando a los campos de mayor homogeneidad.

Dar formato a las fechas

Existen diferentes campos que contienen fecha y otros que contienen hora. A su vez, unos están relacionados con otros, como FINICIO con HALERTA o FFINALIZA con HFINALIZA. Por lo tanto, con el fin de reducir cada vez más el número de campos, se decidió unir esos campos fecha y hora relacionados.

De esta forma, se concatenan ambos campos para luego darles formato GMT, quedando como resultado un único campo que contiene tanto hora como fecha.

Creación de la variable DURALERTA

Durante el análisis inicial de la fase anterior se decidió aprovechar a información que aporta FINICIO y FFINALIZA, y crear un nuevo campo que dé información de la duración de la alerta.

Para ello, se calcula la diferencia entre ambos campos teniendo en cuenta, eso sí, la condición de que existen incidentes que se registran antes de medianoche, pero acaban al día siguiente, aunque se pone la fecha de día anterior. Contado con esto, hay que añadirle 24 horas a la segunda fecha antes de realizar la diferencia. El resultado se expresa en horas.

Normalización de las variables cualitativas

Se trata de una de las variables que mejor representa el ya mencionado problema que existía con la heterogeneidad encontrada en los valores de determinadas variables cuantitativas.

En el caso concreto de la variable SEXO se pudo advertir la existencia de hasta 18 valores diferentes para una variable que en condiciones normales solo debe tener dos posibles valores diferentes: uno para Hombre y otro para Mujer.

Se puede advertir claramente, observando el gráfico de la Figura 4.4 que muchos de estos valores fueron introducidos por error (por las razones previamente estipuladas), pues no se puede extraer información de útil en cuanto al SEXO de prácticamente ninguno de ellos. De la misma forma se observa que estos valores son prácticamente insignificantes y no afectan al resultado global.

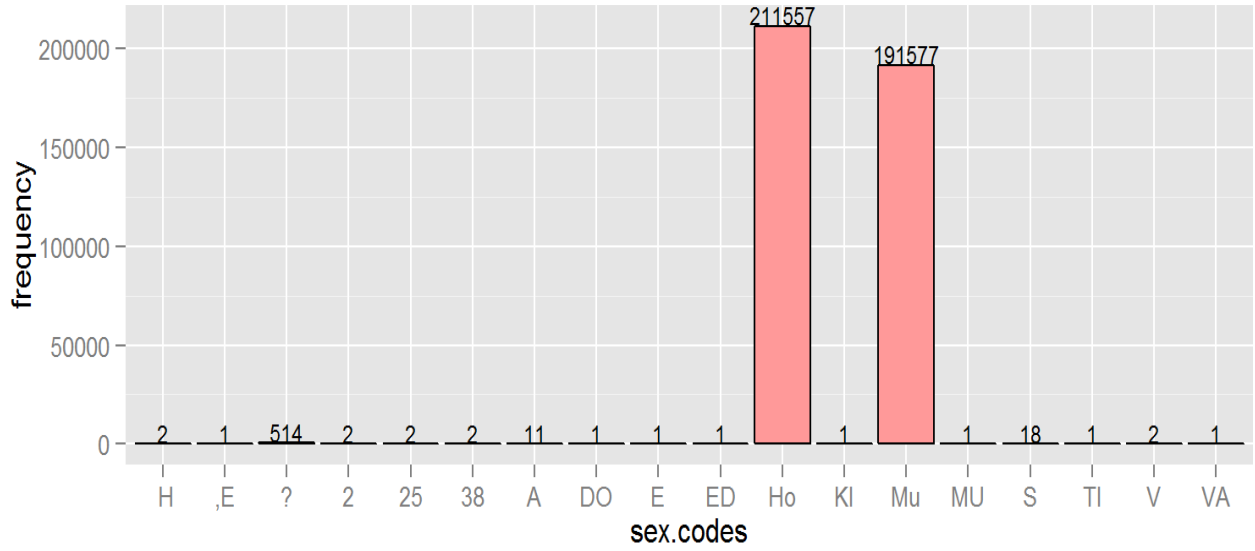


Figura 4.4. Frecuencia absoluta de valores diferentes de la variable SEXO.

Para resolver este problema se decidió dar un valor nulo común a todos los valores que no fueran “Ho” o “Mu”, con el fin de normalizar la variable.

Existen otros casos parecidos con otras variables para las que se sigue el mismo proceso, aunque adaptado a las características de las mismas. Estas otras variables son AMSD y TIDE. En el siguiente gráfico se ve el caso concreto para la variable AMSD:

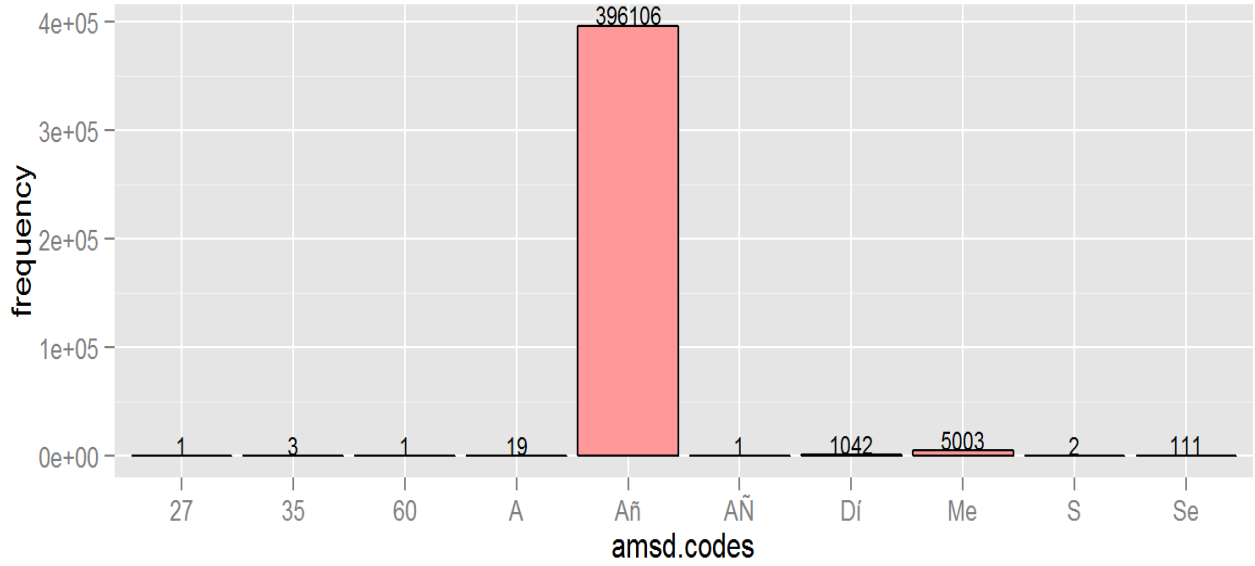


Figura 4.5. Frecuencia absoluta de valores diferentes de la variable AMSD

En este gráfico se ven de nuevo valores sin sentido, que no se consideran significativos, ni alteran el resultado final.

Valores numéricos

Existen varios cambios con valores numéricos como EDAD, CINCIDENTE, etc. que se cargaron como cadenas de caracteres al estar rodeados de dobles comillas tras exportarlos a *.csv*. Para estos casos es necesario convertirlos en tipos numéricos para poder tratarlos como tal durante el procesamiento de los datos en las posteriores fases.

Este paso se realiza rápidamente con una función llamada `as.numeric(x)` que realiza la funcionalidad antes descrita.

Normalización de variables ISLA y MUNICIPIO

Para estas variables existe una singularidad que se debía tener en cuenta. Hasta 2007 cuando un incidente ocurría en alta mar o desde la península por alguna razón, se registraban simplemente como "Mar" o "Península" para ambas variables. Este hecho cambió a partir de ese año donde ya se registraban estos casos con el nombre de la isla al lado para la variable MUNICIPIO. Por ejemplo: "Mar Tenerife".

Teniendo en cuenta esta singularidad surge la necesidad de homogeneizar las variables, igualando ambos periodos. Lo que se decidió fue añadir el nombre de Gran Canaria o Tenerife (según la provincia) a esos casos anteriores a 2007 donde solo venía "Mar" o "Península".

De esta forma se conseguía normalizar estas variables y mantener una serie de valores conocidos, evitando cualquier tipo de inconsistencia.

Variables del tipo *String* a factores

Existen muchas variables del tipo cualitativo, es decir, cuyos contenidos son cadenas de caracteres. Cuando el sistema manipula estos datos lo hace de forma mucho más lenta y costosa que cuando opera con valores numéricos. Es por esto que se asigna a cada cadena de caracteres diferente un valor numérico interno. Es decir, el sistema ve un número, pero nosotros vemos su etiqueta de tipo *String*, siguiendo el sistema clave-valor.

Esta funcionalidad se consigue de manera muy sencilla con un simple comando `as.factor(x, levels)`.

Este proceso se sigue con varias variables como: VALORACION, TIDE, TIPO_RECURSO, NOMTIDE, etc. No se usa con DESCRIP, pues se trata de texto más largo y no tendría sentido realizar esta operación.

Creación de variables AÑO, MES y HORA

Con el fin de poder filtrar nuestros datos desde la interfaz ya sea por años, meses u horas, es necesario extraer estas variables en sus propios campos.

De esta forma se ha procedido a través del campo FINICIO, cogiendo la parte que nos interesaba en caso y convirtiéndola en un valor numérico (AÑO) o en factores (MES y HORA), según correspondiera.

Intersección de datos propios con datos externos del ISTAC

Con el fin de dotar de mayor robustez a los datos que manejábamos, se decidió aportar mayor información que no se encontraba en los datos originales.

Para ello se obtuvo un fichero desde la página web del ISTAC, que poseía información de los municipios (Código, Nombre, Superficie, etc.). De esta forma, tras cargar ese fichero *"info_municipios.txt"*, se unió con nuestros datos usando como clave común el nombre del municipio.

De todos los nuevos campos aportados solo nos quedamos con CODMUN (Código del municipio). Esta variable que identifica a cada municipio con un código permitirá manipular los datos de forma mucho más rápida (usando valores numéricos y no nombres).

Almacenamiento y agrupación

Finalmente se procede a guardar los datos ya limpios en archivos *.RData*. Este formato de archivo es propio de *R* y se almacenan de tal forma que ocupan un espacio muy pequeño en comparación a su peso original. Por ejemplo, si los archivos originales ocupaban 200 Mb cada uno, una vez compresos en este formato ocupan solamente 13 Mb de media, una reducción casi de más del **90%** en espacio.

Posteriormente, se procede a unir todos los archivos individuales en un único fichero global, que servirá como punto de partida para toda la manipulación y el procesamiento que se procederá a realizar a continuación. Este archivo final se cribará para quitar variables que no se usarán por no

considerarlas suficientemente relevantes para nuestro proyecto. Entre estas se encuentran: FFINZALIZA, HFINALIZA, MUNICIPIO, TIPO_RECURSO, CINTERVENCION, CODIGO_FINALIZACION_RECURSO, etc.

Un paso final importante en este proceso es el hecho de quedarse con la menor cantidad de nombres posible en los valores de los campos. Lo ideal era tener datos representados por valores numéricos y códigos. De esta forma, se almacenaron diferentes archivos *.RData* con equivalencias de clave-valor para variables como TIDE y NOMTIDE, TRECURSO y TIPO_RECURSO, etc.

Así, se consultan estas tablas "hash" en tiempo de ejecución y se obtiene muestra por pantalla el nombre y no el código al usuario, mientras que el sistema realmente trabaja con los valores numéricos o códigos.

4.4 Análisis y diseño inicial de la interfaz gráfica

Durante este apartado se describirán las consideraciones y reflexiones que se realizaron para idear el aspecto final de la interfaz gráfica a lo largo de las diferentes reuniones.

Desde una de las primeras reuniones donde se definió el primer objetivo a alcanzar, hasta las últimas donde se estableció los últimos detalles a incluir, se fue definiendo poco a poco las diferentes características de la aplicación gráfica.

A continuación se listan las diferentes pestañas de la aplicación, así como la evolución en las decisiones con el transcurso de las reuniones:

Mapa de Canarias

El primer objetivo que se estableció fue mostrar un mapa de Canarias, dividido en municipios. Este poseería las siguientes características:

- Existiría una escala de **10 colores**: azul, verde, amarillo, naranja, rojo (dos tonos para cada uno). En las reuniones iniciales, se usaba una escala de un solo color (desde blanco hasta el color).
- Esta escala representaría un **rango de porcentajes** (por ejemplo [10%-12%]). Inicialmente la escala representaba valores absolutos acumulados por cada municipio, no porcentajes.

- El usuario podría filtrar los datos por **tipo de incidente, valoración y recurso** empleado en el incidente.
- Cada municipio tendría un valor que vendría dado por el porcentaje de incidentes (según selecciones del usuario) de ese municipio con respecto al total de incidentes de toda Canarias. Es decir, si Adeje tiene un valor de 12% quiere decir que, para el total de incidentes seleccionado, Adeje representa ese porcentaje respecto al resto de municipios canarios.
- De esta forma, los municipios se colorearían de diferentes colores según se encuentren en el rango de porcentajes correspondiente.
- Además, **se filtrará por año**. Es decir, para cada año los valores cambian (así como los rangos de porcentajes).
- Posteriormente, surgió la idea de crear un ***slider*** a lo largo de los años. Esta funcionalidad permitiría darle a un botón de *play* y ver como el mapa va cambiando año a año, con un intervalo ajustable entre cada transición.

Como una de las peticiones finales, se añadió la funcionalidad de exportar el mapa a diferentes formatos (PDF, JPG, PNG, etc.), pues es resultaba muy útil para obtener informes de forma inmediata ante cualquier comunicado o rueda de prensa inminente.

Gráficas lineales interactivas

En un punto intermedio de las reuniones, tras visualizar las primeras versiones del mapa, se decidió que éste daba una gran visión global de los datos, pero que si se quería ahondar en variables concretas para algún municipio en concreto y añadir más "granularidad" a los resultados era necesario crear una pestaña de gráficas interactivas.

Esta pestaña incluiría las siguientes características:

- A la izquierda habría un **panel de control**, y a su derecha, la visualización de las **gráficas**.
- El panel de control serviría para seleccionar diferentes entradas y que a su vez éstas se reflejaran en las gráficas.
- En el panel de control se podrá seleccionar la **isla** deseada (o toda Canarias) y el **municipio** de la isla (o toda la isla).

- Además, el panel de control incluirá una selección de las variables que ocuparán cada uno de los ejes de la gráfica.
- En el eje X, la variable temporal podrá variar entre los **años**, los **meses** o las **horas**.
- En el eje de las Y, se seleccionará la variable a medir: **tipo de incidente, valoración y tipo de recurso utilizado**. Inicialmente se representarían únicamente el top-6 de valores con más incidentes en ese periodo temporal para esa localización. Posteriormente, se decidió habilitar la posibilidad de seleccionar un valor en concreto (por ejemplo, el tipo de incidente "Enfermedad")y que apareciera solo en la gráfica.
- Los valores representados serían los **promedios** según la variable temporal elegida. Por ejemplo, si se elige "meses", se calcularía el valor promedio para ese mes de cada uno de los 10 años para la variable del eje Y elegida.
- Como ejemplo podríamos seleccionar para la isla de Tenerife, en el municipio de Adeje y ver como se distribuye el tipo de recurso utilizado "Policía Nacional" a lo largo de las horas del día durante los 10 años de datos.

Gráficas de barras interactivas

En las últimas reuniones se decidió aumentar el nivel de detalle en información y añadir una pestaña de gráficas de barras con valores porcentuales. Éstas permitirían comprobar cómo se distribuyen determinadas variables por las islas en términos porcentuales.

Esta pestaña incluiría las siguientes características:

- A la izquierda habría un panel **de control**, y a su derecha, la visualización de las **gráficas**.
- El panel de control serviría para seleccionar diferentes entradas y que a su vez éstas se reflejaran en las gráficas.
- En el panel de control se podrá seleccionar una variable temporal concreta (**año, mes** o **hora**). Luego, se debería seleccionar que valor de esa variable se elige (qué año, ejemplo).
- En el eje X, se situarán las diferentes **islas**.

- En el eje de las Y, se visualizará el valor porcentual de cada isla para la variable elegida (**tipo de recurso, valoración** y **recurso utilizado**). Se usará una barra para cada valor de la variable elegida (se utilizará solo el top-6 por razones de espacio), teniendo así cada isla sobre sí seis barras.
- Por ejemplo, se podría seleccionar la variable temporal "año" con el valor 2010. También seleccionamos la variable "tipo de incidente". En este caso se desplegaría una gráfica con todas las islas en el eje X. Sobre cada isla se levantarán 6 barras (una para cada variable del top-6 de "tipo de incidente" - Bomberos, Guardia Civil, etc.-). De esta forma con cada barra vemos el porcentaje que representa esa variable en esa isla para año, respecto al total de incidentes de Canarias (si sumamos los porcentajes de cada isla para un valor concreto ("Guardia Civil" , por ejemplo), sumaríamos el 100%, obviamente.

Nube de Tags

Durante todas las reuniones siempre se había comentado la idea de poder realizar algo interesante con la variable **DESCRIP**, que contenía una descripción adicional realizada en texto libre por el médico que atendiera la incidencia.

En un principio, se pensó que se trataba de textos que no guardarían relación unos con otros, aún siquiera para incidentes similares. Sin embargo, en una de las últimas iteraciones de la etapa de análisis de los datos se pudo ver que, al contrario de lo que pensaba, el texto de esta variable estaba bastante estructurado. De hecho, había muchas sentencias que se repetían continuamente, como por ejemplo, "Respira con dificultad" o "Precisa atención médica".

Estos pequeños "descubrimientos" nos llevaron hacia una idea concreta. La posibilidad de crear una **nube de términos**. Podríamos decir que una nube de palabras o nube de etiquetas es un recurso visual que se utiliza para representar las palabras más destacadas que componen un determinado texto.

Estas nubes de palabras suelen presentarse a modo de figura abstracta, en las que son representadas de un mayor tamaño aquellas palabras que aparecen con más frecuencia o son más importantes.

Por lo tanto, nuestro caso se ajustaba perfectamente, tenemos un texto (todas las descripciones) y usaríamos las palabras para crear la nube.

La pestaña a crear poseería las siguientes características:

- Se crearía un panel de control a la izquierda, con la nube de palabras a la derecha.
- En el panel de control encontraríamos dos posibles controles del usuario: el número mínimo de ocurrencias de una palabra para aparecer en la nube y el número máximo de palabras que puede tener la nube.
- La nube se iría ajustando de forma reactiva según se cambien los valores de las selecciones del usuario.

4.5 Procesamiento de datos

Esta sección pretende describir todos los pasos seguidos desde que se obtuvo el archivo único con la tabla de datos de los 10 años hasta crear los diferentes subconjuntos que se cargan en tiempo de ejecución para el mapa, las gráficas y la nube de términos (el código completo se encuentra en el apéndice). Este procesamiento previo permite dotar de mayor rapidez y versatilidad a la herramienta haciendo que la interfaz maneja con la máxima fluidez teniendo en cuenta, que se manipulan más de 7 millones de registros.

4.5.1 Uso de librerías *Big Data*

A la hora de cargar tal cantidad de registros en una tabla, resultaba realmente lento (más de 1 minuto), por lo que se decidió buscar alternativas.

Para resolver este problema de la carga y tratamiento de datos utilizamos la librería de *R, "ffbase"*. Este paquete nos permite trabajar con objetos y estructuras de datos masivas (por ejemplo, datos de secuenciación genómica).

Ffbase utiliza métodos de acceso rápidoa los datos (mediante índices y paginación en memoria) y nos permite crear una estructura de datos óptima para trabajar con ella. A continuación se puede apreciar una carga de datos con esta librería:

```
library(ffbase)
start <- proc.time()
datos_global <- load.ffdf(dir="data_ffdf")
end<-proc.time() - start
## [1] "Estructura ffdf: 0.13 segundos"
```

Figura 4.6. Ejemplo de carga de datos con el paquete *ffbase*.

La Tabla 4.1 contiene mediciones de carga de datos con la librería *ffbase* y otras con el sistema tradicional de tablas *data.frame*:

Tiempo de carga en *data.frame* (segundos)	Tiempo de carga con *ffbase* (segundos)
66.81	0.06
68.02	0.16
62.51	0.17
57.35	0.25
58.95	0.09
60.01	0.11
64.33	0.19

Tabla 4.1. Tabla de tiempos de carga.

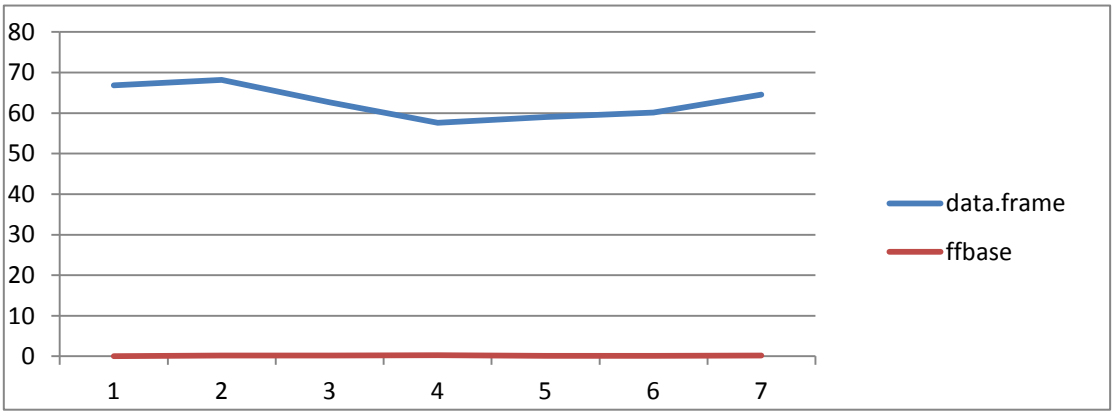


Figura 4.7. Gráfica de tiempos de carga.

Se evidencia en la Figura 4.7. una **reducción casi del 100%** en el tiempo de carga. Algo increíble, que habla muy bien de las ventajas de esta librería.

Además, el espacio que ocupa en RAM es de **5.3 Mb** de media respecto a los **89 Mb** del método antiguo.

4.5.2 Datos para el Mapa de Canarias

Los datos que dan soporte al mapa interactivo se manipulan casi totalmente en tiempo de ejecución. Es ahí cuando se calculan los porcentajes, los rangos para los colores, se filtran por variables, etc.

Antes de ello, eso sí, se hace un pequeño procesamiento previo. Se realiza una agregación de los registros juntando estas variables:

- **AÑO:** permitirá filtrar por año en el mapa
- **CODMUN:** conecta cada municipio del mapa con los datos
- **TIDE:** permitirá la selección del tipo de incidente
- **VALORACION:** se usará para la selección de la valoración del incidente
- **TRECURSO:** permitirá seleccionar el tipo de recurso usado

Como resultado de esta agregación, cuentan los registros que tienen los mismos valores para las variables anteriores, creando una nueva variable **N** que almacena el total de esa cuenta. El resultado se aprecia en la Figura 4.8:

	ANYO	TIDE	VALORACION	TRECURSO	CODMUN	N
43532	2006	SES7	Leve	MOT	38023	1
43533	2006	SES7	Leve	MOT	38037	1
43534	2006	SES7	Leve	PGC	35002	5
43535	2006	SES7	Leve	PGC	35003	1
43536	2006	SES7	Leve	PGC	35004	1
43537	2006	SES7	Leve	PGC	35005	1
43538	2006	SES7	Leve	PGC	35006	7
43539	2006	SES7	Leve	PGC	35009	2
43540	2006	SES7	Leve	PGC	35011	3
43541	2006	SES7	Leve	PGC	35012	2
43542	2006	SES7	Leve	PGC	35013	1
43543	2006	SES7	Leve	PGC	35014	1
43544	2006	SES7	Leve	PGC	35015	1
43545	2006	SES7	Leve	PGC	35019	1
43546	2006	SES7	Leve	PGC	35021	1

Showing 43,532 to 43,546 of 268,553 entries

Figura 4.8. Datos agregados para el mapa interactivo

4.5.3 Datos para las Gráficas lineales y de barras interactivas

Estos datos, al contrario que para el mapa interactivo, sí que requieren un gran trabajo previo con el fin de satisfacer todos los posibles casos para diferentes selecciones del usuario en las gráficas interactivas. A pesar de esto, una gran carga del trabajo se realiza aún en tiempo de ejecución.

Este procesamiento para gráficas lineales se divide en dos etapas:

- Preparación de bancos de datos completos
- Creación de subconjuntos agregados

Durante la **primera etapa**, se crearon diferentes tablas de datos que contenían datos completos de las diferentes agrupaciones de variables que intervenían en esa pestaña de la interfaz. Por ejemplo, se crea una tabla con el producto escalar de estas variables, es decir, todas las combinaciones posibles de dichas variables (excepto municipios con islas, pues no todos los municipios están en todas las islas. Se tiene en cuenta esa condición).

En la Figura 4.9 se ve un ejemplo de este caso. Para cualquier municipio de una isla cualquier tipo de incidente en cualquier mes del año:

	CODMUN	MES	TIDE	ISLA
1	35028	enero	SEIC	Lanzarote
2	35019	enero	SEIC	Gran Canaria
3	35012	enero	SEIC	Gran Canaria
4	35017	enero	SEIC	Fuerteventura
5	35022	enero	SEIC	Gran Canaria
6	35016	enero	SEIC	Gran Canaria
7	35027	enero	SEIC	Gran Canaria
8	35024	enero	SEIC	Lanzarote
9	35006	enero	SEIC	Gran Canaria
10	35026	enero	SEIC	Gran Canaria
11	35004	enero	SEIC	Lanzarote
12	35011	enero	SEIC	Gran Canaria
13	35023	enero	SEIC	Gran Canaria
14	35003	enero	SEIC	Fuerteventura
15	35002	enero	SEIC	Gran Canaria

Showing 1 to 15 of 37,632 entries

Figura 4.9. Muestra de un banco de datos completo

Este proceso es realmente necesario, pues se usa para rellenar los huecos que existen en los datos reales. Es decir, cuando se agregan datos existe la posibilidad de que para ciertas combinaciones de variables no existan registros. Por ejemplo, es posible que en el municipio de "Valverde" (El Hierro) no exista ningún registro de "Fiebre" con Valoración "Grave" a las "2:00" de la mañana de cualquier mes de cualquier año. Por eso se crean estos bancos de datos para rellenar estos huecos de tal forma que se reflejen en las gráficas (de otra forma se comportarían de manera inexacta, uniendo puntos que no son contiguos, etc., produciendo así resultados erróneos).

Durante la **segunda etapa**, se procedió a crear los diferentes subconjuntos que se cargan durante la ejecución de la aplicación interactiva.

Este proceso consta de varios pasos. El primero es realizar una agrupación sobre el archivo total de registros usando como claves las variables necesarias según el caso que podría seleccionar el usuario.

Por ejemplo, si se busca información del tipo de recurso en un municipio a una hora concreta, es preciso agrupar estar variables, además del mes y el año (para este caso en concreto), lo que se explicará a continuación. Una vez se agrupan estas variables, se cuentan los registros para las T-uplas iguales, generando la nueva variable **N**.

El siguiente paso consiste en hacer una agrupación parecida sobre el conjunto resultante del anterior paso, pero solo cogiendo de las variables temporales la que realmente voy a usar (horas en nuestro ejemplo anterior). La principal diferencia es que ahora al realizar la agrupación se realiza un promedio de los registros iguales con `mean(x)`. De esta forma, para nuestro ejemplo anterior, nos quedaríamos con el promedio de incidentes cada hora a lo largo de cada mes durante los 10 años de datos. Este valor se redondea a las décimas para facilitar la visualización.

	ISLA	CODMUN	HORA	TRECURSO	N
1	El Hierro	38013	00:00	BBO	1.4
2	El Hierro	38013	00:00	CCA	1.0
3	El Hierro	38013	00:00	CFA	1.0
4	El Hierro	38013	00:00	CGI	1.3
5	El Hierro	38013	00:00	CJE	1.0
6	El Hierro	38013	00:00	COT	1.0
7	El Hierro	38013	00:00	CSE	1.0
8	El Hierro	38013	00:00	CTU	1.0
9	El Hierro	38013	00:00	MCR	1.0
10	El Hierro	38013	00:00	MGT	1.0
11	El Hierro	38013	00:00	MHR	1.0
12	El Hierro	38013	00:00	MMA	1.0
13	El Hierro	38013	00:00	MOT	1.4
14	El Hierro	38013	00:00	MPC	1.0
15	El Hierro	38013	00:00	MSV	1.2

Showing 1 to 15 of 144,983 entries

Figura 4.10. Muestra de ejemplo de subconjunto de datos para gráficas

Finalmente, se realiza una unión completa (*full join*) con el banco de datos completos correspondiente. Como resultado final de nuestro ejemplo quedaría un subconjunto de datos que permitiría filtrar por municipio y tipo de recurso, mostrando en la gráfica los valores promedios de esos recursos a lo

largo de las horas del día. En la Figura Figura 4.10 se ve el resultado del ejemplo seguido.

Para las gráficas de barras se sigue el mismo proceso, pero intercambiando un cálculo promedio, con un cálculo porcentual.

4.5.4 Nube de *tags*

Para la nube de palabras solo es necesario un pequeño procesamiento previo. Éste consiste en extraer la variable **DESCRIP** de todos los registros para posteriormente concatenarlos en cadena de caracteres de gran tamaño, eliminando signos de puntuación. y dejando cada palabra separada por espacios.

4.6 Implementación de la interfaz gráfica interactiva

En este capítulo se describen los pasos a seguir para construir la aplicación web interactiva que permite manipular los datos y obtener resultados de forma reactiva.

Como se ha establecido anteriormente, para la implementación de la interfaz gráfica se ha decidido utilizar el paquete *Shiny*, que aporta herramientas para crear una aplicación web interactiva utilizando el lenguaje *R* y vinculando de forma directa los datos cargados en el entorno.

4.6.1 ui.R, server.R y global.R

Para realizar una aplicación con *Shiny* son indispensables dos archivos y existe otro opcional: ui.R, server.R y global.R.

ui.R

Este fichero es el que (como su nombre indica) permite crear la interfaz del usuario, es decir, la parte visual. Para ello, es aquí donde se crea la estructura externa de la aplicación: vistas, pestañas, botones, listas desplegables, etc.

server.R

Desde este archivo se controla toda la lógica interna. El código que figura en este fichero gestiona el comportamiento de botones, selecciones en listas, desplazamientos en *sliders*, etc. Además, es donde se debe manipular y procesar los datos para adaptar los resultados a la selecciones del usuario en tiempo de ejecución. Desde aquí se realizan las llamadas a funciones para el mapa, las gráficas, la nube de tags, etc.

global.R

Por último, existe un archivo opcional encargado de manejar variables a nivel global. En este fichero se cargaran las librerías y paquetes necesarios, además de alguna otra variable que afecte a los otros dos archivos de igual forma o que vaya a permanecer inmutable durante la aplicación (constantes).

A continuación, se procederá a describir los pasos seguidos para la implementación de cada una de las pestañas y elementos de la interfaz, incluyendo cualquier desarrollo previo, así como el código usado para cada uno de estos elementos en los archivos anteriormente descritos (todo el código descrito a continuación se encuentra en el Apéndice).

4.6.2 Mapa de Canarias

Para la realización de este mapa interactivo existió la necesidad de realizar un desarrollo y preparación previas, pues como es lógico no existía un archivo *GeoJSON* de Canarias.

Un archivo *GeoJSON* es un formato estándar abierto diseñado para representar elementos geográficos sencillos y ligeros, junto con sus atributos no espaciales, basado en *JavaScript Object Notation*. El formato es ampliamente utilizado en aplicaciones de cartografía en entornos web al permitir el intercambio de datos de manera rápida, ligera y sencilla. Permite relacionar a cada elemento del mapa (países, regiones, municipios, etc.) con un código identificativo que se usará luego desde el código de la aplicación en que se implemente.

Para la realización de este mapa fue necesario descargar, primero, el archivo *ShapeFile* del mapa de Canarias. Un fihero de tipo *ShapeFile* es un

formato vectorial de almacenamiento digital donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos.

Usando el *QGIS*, un software que permite cargar un mapa en formato *ShapeFile* y crear el fichero *GeoJSON* con el cual identificar a cada municipio con su código (CODMUN) asociado. En la Figura 4.11 se puede apreciar el software durante este proceso:

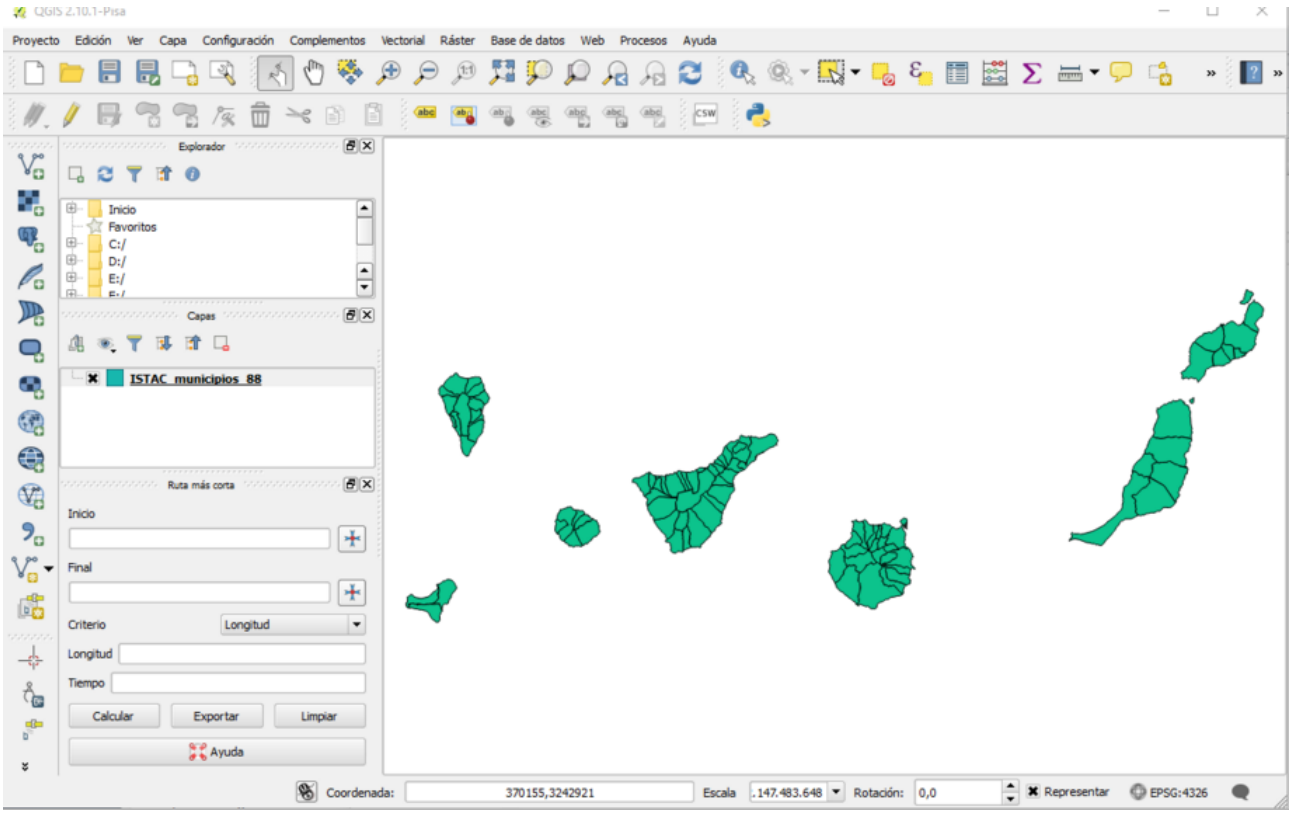


Figura 4.11. Creación del archivo *GeoJSON* a partir de *ShapeFile* con *QGIS*.

Una vez obtenido el mapa en formato *GeoJSON*, éste ya se integrar usar con la librería *Javascript Highmaps*. Esta librería permite crear mapas que interactúan con datos usando ese lenguaje.

En el caso particular que nos ocupa, existe un paquete de *R*, *highchartsUtils*, que hace la función de *wrapper* entre el lenguaje que usamos y *JavaScript*. De esta forma, podemos crear el mapa de Canarias, conectarlo con nuestros datos y adaptarlo a nuestras necesidades usando las funcionalidades que nos aporta esta librería. En la Figura 4.12 se puede observar un ejemplo de un mapa de Alemania usando *Highmaps*.

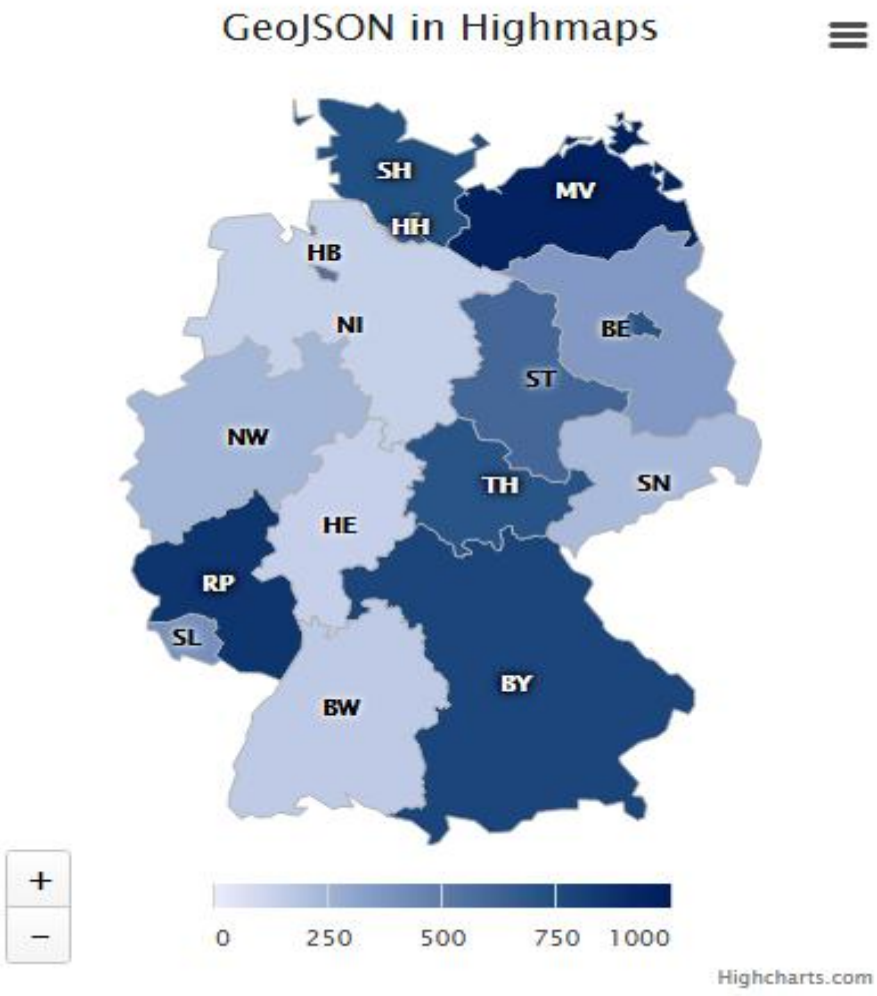


Figura 4.12. Mapa de Alemania usando librería *Highmaps*.

Una vez se ha creado el mapa, se describe la implementación llevada a cabo en cada fichero necesario para la aplicación:

ui.R

El primer paso es crear la parte visual del mapa. Al ser el primer elemento de todos los creados, se creará, de la misma forma, la estructura de la aplicación en sí.

Primero, hay que usar la función `shinyUI` para crear la vista, definiendo el tipo de la misma como `fluidPage`. Esto permite organizar los elementos en columnas y filas "invisibles" generando un orden interno.

A continuación, se le da nombre a la aplicación y se establece el estilo de pestañas que se desea. La primera pestaña de nuestra aplicación será la que muestre el "Mapa". Dentro de esta pestaña habrán dos elementos:

- **el panel principal (`mainPanel`):** este panel será donde se aloje el mapa en sí. Se situará la izquierda de la vista.
- **el panel de control:** se trata de un `absolutePanel` que contiene la descripción del proyecto, además de otro `absolutePanel` embebido que aloja los controles de usuario.

Este panel se configura para que sea móvil (situarlo donde mejor convenga al usuario). Además, éste incluirá tres listas desplegables que permiten filtrar los resultados que muestra el mapa (**tipo de incidente, valoración y tipo de recurso**).

Por último, el panel incluye un *slider* que controla el año (2005-2014) al que pertenecen los resultados que se muestran. Como función adicional existe un botón de *play* que permite pasar de un año a otro con un intervalo de segundos ajustable, mientras el mapa se actualiza en consecuencia.

server.R

En este fichero se describirá el procesamiento y filtrado de datos que se hace en tiempo de ejecución.

El primer paso es crear la función que permite interactuar con los elementos de ui.R, `shinyServer (function (input, output))`. Esos parámetros *input* y *output* son lo conectan con esos elementos a través de identificadores únicos.

Para poder filtrar los datos según las selecciones del usuario, primero se recoge el valor de esas variables (como el usuario selecciona nombres, es preciso convertir esos nombres a códigos a través de las tablas *hash* clave-valor creadas en capítulos anteriores.

Posteriormente, se crean todas las posibles combinaciones de selecciones del usuario posibles, pues lo datos variarán en consecuencia. Una vez se entra en la condición correspondiente, se filtran los datos usando funciones de la librería *dplyr:*

- Con la función `filter` se hace un filtrado de los datos en base al valor de las variables seccionadas por el usuario.
- Usando la función `ddply`, podemos agregar los datos de tal forma que se agrupen los registros que tengan los mismos valores para las variables

seleccionadas en la agrupación, sumando el número de incidentes de cada registro agregado con esas claves (**N**).

- Finalmente, con la función `prop.table` se atribuye al valor total de incidentes (**N**) de cada registro del subconjunto ya agregado, un valor porcentual respecto al total de los mismos.

El siguiente paso es crear el intervalo de clases usando `ClassIntervals`. Esta función permite crear rangos numéricos basados en los datos que le pasan por parámetro y, además, estos rangos están basados en la distribución de esos valores de entrada (es decir, si la mayoría de los valores se concentran en números pequeños, se crearan rangos más pequeños, y para los valores altos más dispersos, rangos más grandes).

Por último, se asignan los colores a esos rangos y se llama la función principal `highmapsChoropleth`, la cual es la que crea el mapa en sí, basándose en los datos de entrada ya procesados, indicando qué variable representar y qué variable usar como clave para el archivo de mapa *GeoJSON*.

El propio panel del mapa posee, además, una serie de características propias:

- al situar el puntero sobre algún municipio, este cambia su color y despliega información sobre el rango de porcentajes al que pertenece y el nombre del mismo.
- Bajo el mapa se encuentra la leyenda de colores y rangos de porcentajes. Clicando sobre cada uno, habilitamos o deshabilitamos ese color del mapa.
- Además, en la esquina superior derecha existe un menú contextual que permite exportar el mapa en diferentes formatos (*PDF, jpg, png,* etc.).

A continuación en la Figura 4.13 se muestra la interfaz acabada:

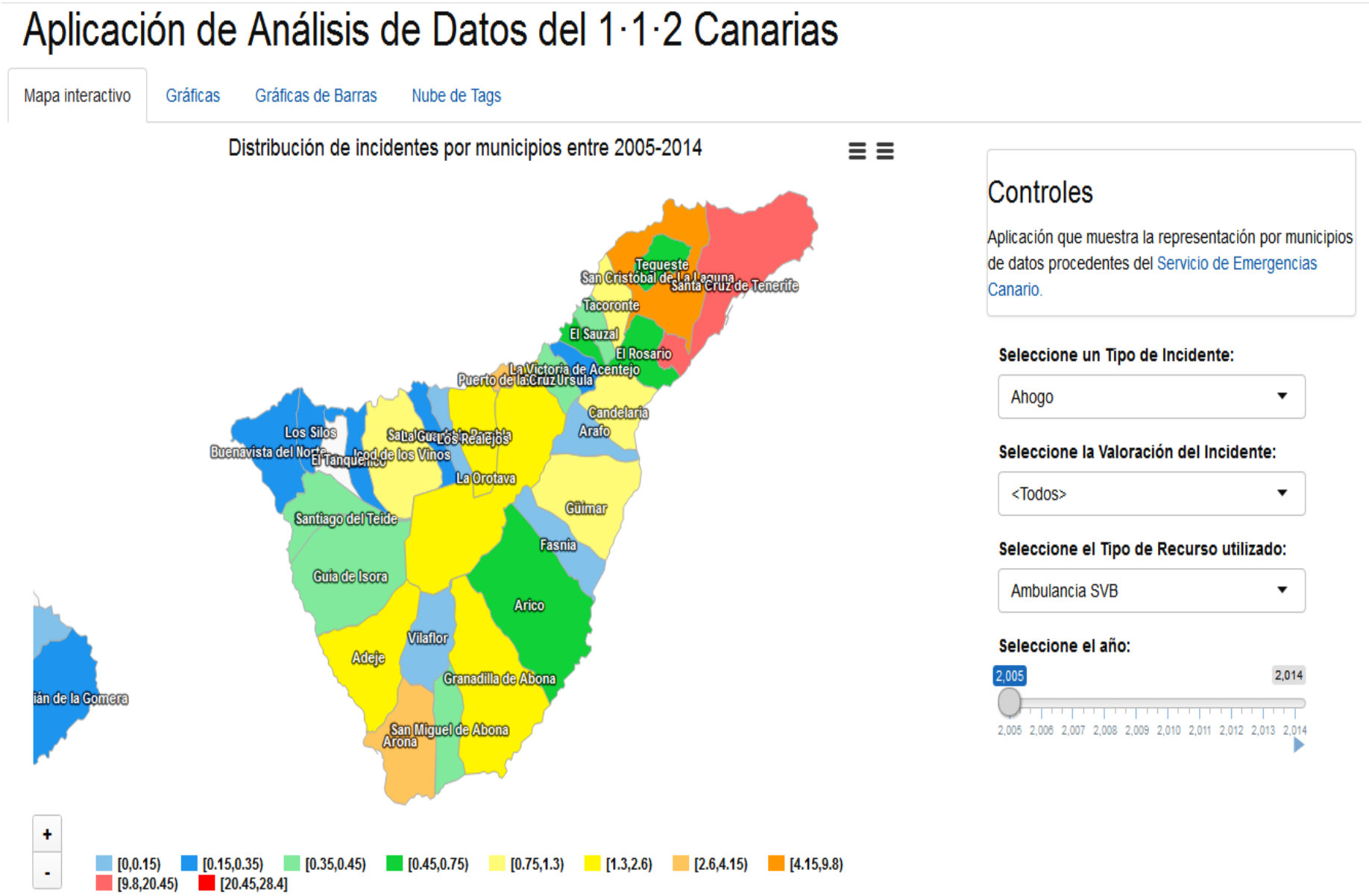


Figura 4.13. Mapa interactivo de Canarias.

4.6.3 Gráficas lineales interactivas

Para la creación de las gráficas se utilizo la librería *rCharts*. Se trata de un paquete que permite crear, personalizar y publicar visualizaciones interactivas en *JavaScript* desde *R*. Para nuestro caso particuar usaremos un tipo de gráfico específico, *HighCharts*. Este tipo de gráfico lineal se adapta a nuestros datos y a los resultados que queremos mostrar. En la Figura 4.14 se aprecia un ejemplo de este tipo de gráficas:

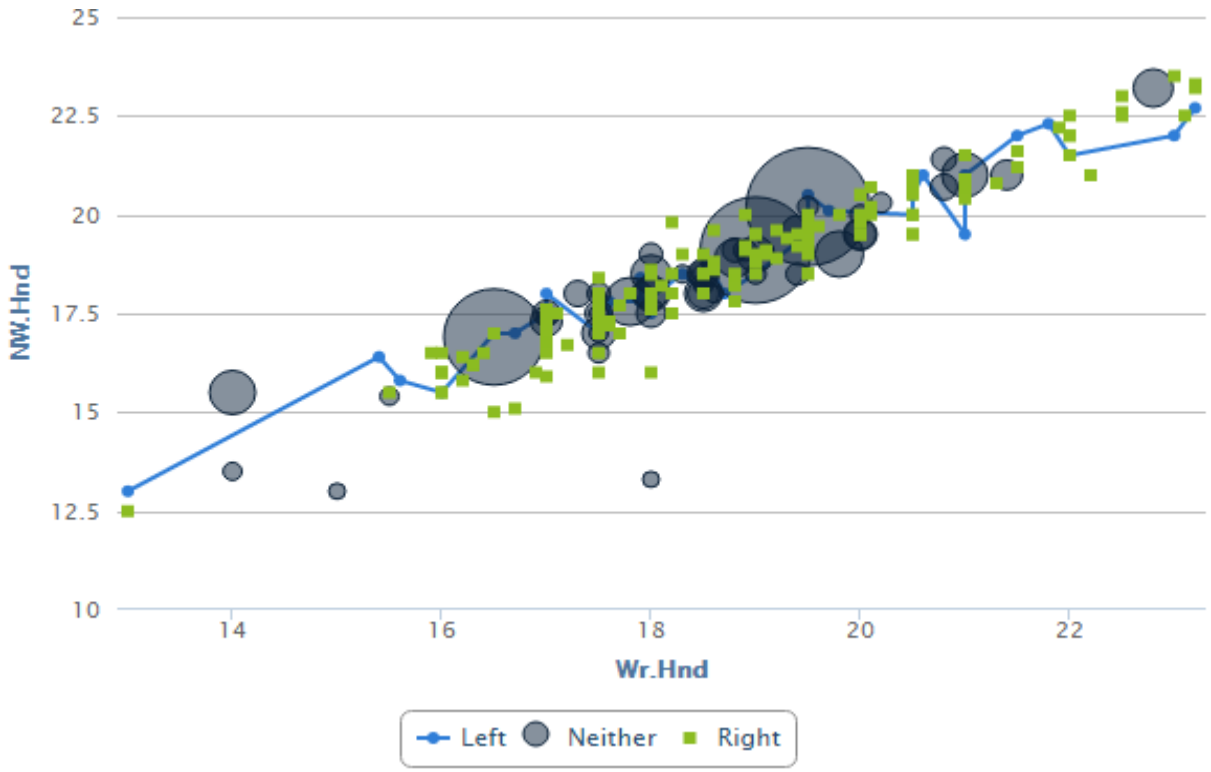


Figura 4.14. Ejemplo de gráfica *HighCharts* de *rCharts*.

A continuación, se describirá el proceso de implementación de estas gráficas para cada archivo de *Shiny:*

ui.R

La pestaña de las gráficas lineales es la más compleja de todas, pues existen muchas selecciones posibles para el usuario; además de que la propia interfaz de control cambia con según qué selecciones.

Para esta vista usamos el tipo `sidebarLayout`, que permite crear una barra de control fija a la izquierda, mientras que la gráfica se despliega a la derecha.

El **panel de control** tiene la siguiente estructura:

- Primero, existe una selección de **isla** (por defecto, se le da el valor "Todas" para Canarias en general).
- Si se ha elegido una isla, aparece un nueva selección, en este caso de **municipio**. Para esto, se ha usado un `conditionalPanel`, que permite habilitar elementos visuales dependiendo de alguna condición. Es importante que solamente se muestren los municipios de cada isla (no todos, para evitar confusiones). Esto se consigue a través una pequeña consulta a otra tabla de soporte (*isla.codmun*) que relaciona a cada municipio con su isla. Por defecto, el valor seleccionado es "Todos" para la isla escogida en general.
- La siguiente selección es elegir la **variable temporal** a mostrar en el eje X de la gráfica. Existen tres `radioButtons` a seleccionar (Año, Mes y Hora).
- A continuación se vuelven a desplegar tres `radioButtons` para seleccionar la variable a medir (**tipo de incidente**, **valoración** y **tipo de recurso** utilizado) en el eje Y.
- Si se selecciona "tipo de incidente" o "tipo de recurso" aparecerá una lista desplegable, gracias a otro `conditionalPanel`, en la que se podrá elegir el valor a mostrar en la gráfica (por defecto, se muestra el top-6 de valores con más incidentes).

server.R

En este fichero se describirá el procesamiento y filtrado de datos que se hace en tiempo de ejecución para las gráficas.

El primer paso es obtener los valores de las variables según las selecciones que ha realizado el usuario. Luego, se consultan las tablas de soporte para intercambiar las variables de tipo nombre que selecciona el usuario, por su variable código equivalente (municipio por su código, nombre del tipo de incidente por su código, etc.).

Posteriormente, se implementan los **condicionales** que cubren todas las combinaciones de diferentes selecciones posibles del usuario (de esta forma se carga el subconjunto de datos -correspondiente a la selección del usuario- creado para ese propósito en los capítulos anteriores).

Una vez se entra en la condición correspondiente, por ejemplo, se ha seleccionado la variable "tipo de incidente" para la variable temporal "Años", eligiendo una isla y municipios concretos. Primero, se debe comprobar si el valor de la variable "tipo de incidente" es "Top-6" o una variable concreta ("Fiebre", por ejemplo):

- si es **"Top-6"**, se hará un filtrado del subconjunto de datos cargado (el que corresponda según el condicional en el que se entró) según el municipio elegido. Posteriormente, realizamos un listado de cada valor de la variable con el número de incidentes que suma cada uno (`tapply(x$N, x$TIDE, sum,na.rm = TRUE)`). De todos, cogemos solamente los nombres de los primeros 6.

  A continuación, realizamos un nuevo filtrado del subconjunto original para quedarnos únicamente con los registros que corresponden a variables del top-6 y de ese municipio.

  Por último, con la función `HighCharts`, creamos cada una de las series para cada valor del top-6. Para cada serie se le da el nombre del valor ("Enfermedad", por ejemplo), la variable a medir (**N**) y (muy importante) el parámetro `connectNulls`. Deshabilitando este parámetro se dejan huecos en las gráficas para los datos que no existen (de otra forma uniría los puntos creando falsos resultados).

- si se seleccionó un **valor concreto** (por ejemplo, "Ahogo"), simplemente se filtran el subconjunto de datos correspondiente por el municipio y ese valor de tipo de incidente elegido.

  A continuación, se usa la función `HighCharts` para crear una única serie para ese valor, poniéndole su nombre, su total de incidentes (**N**) y el parámetro `connectNulls` deshabilitado.

Cabe destacar varias características de las propias gráficas interactivas:

- Bajo la gráfica se encuentra la leyenda de colores y valores de la variable. Clicando sobre cada elemento se habilitan o deshabilitan la serie de ese valor.
- Cuando se deshabilita una serie, los rangos del eje Y se redimensionan para adaptarse a los valores numéricos de las series restantes.

A continuación en la Figura 4.15 se muestra la interfaz acabada:

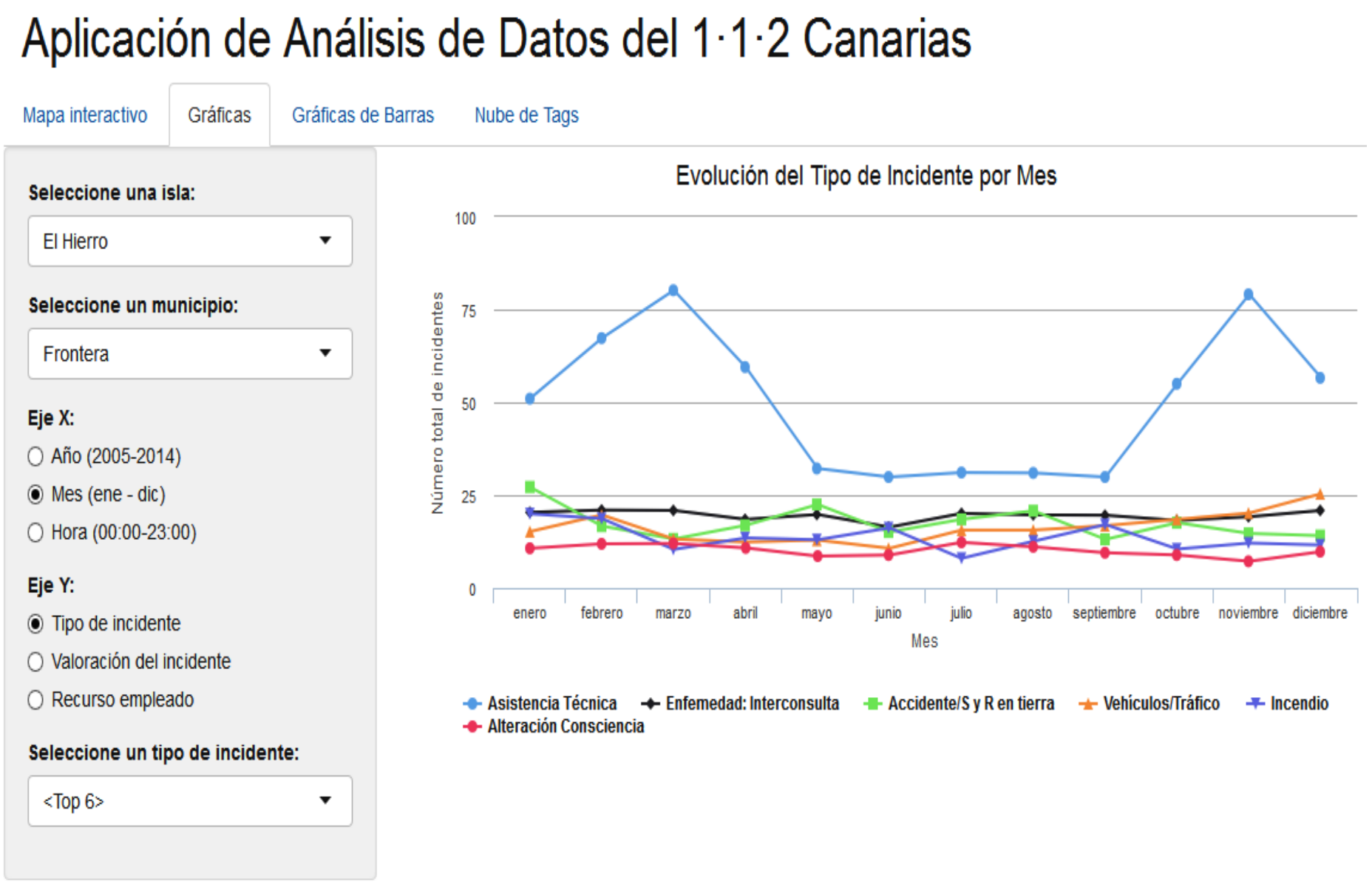


Figura 4.15. Gráfica lineal interactiva.

4.6.4 Gráficas de barras interactivas

Para la creación de estas variables se uso de nuevo la librería *rCharts*, pero usando el tipo de gráficas *NVD3*. Estás gráficas de barras tienen la particularidad de ser interactivas, pudiendo mostrarse de la forma normal o apiladas. En la Figura 4.16 se puede observar un ejemplo:

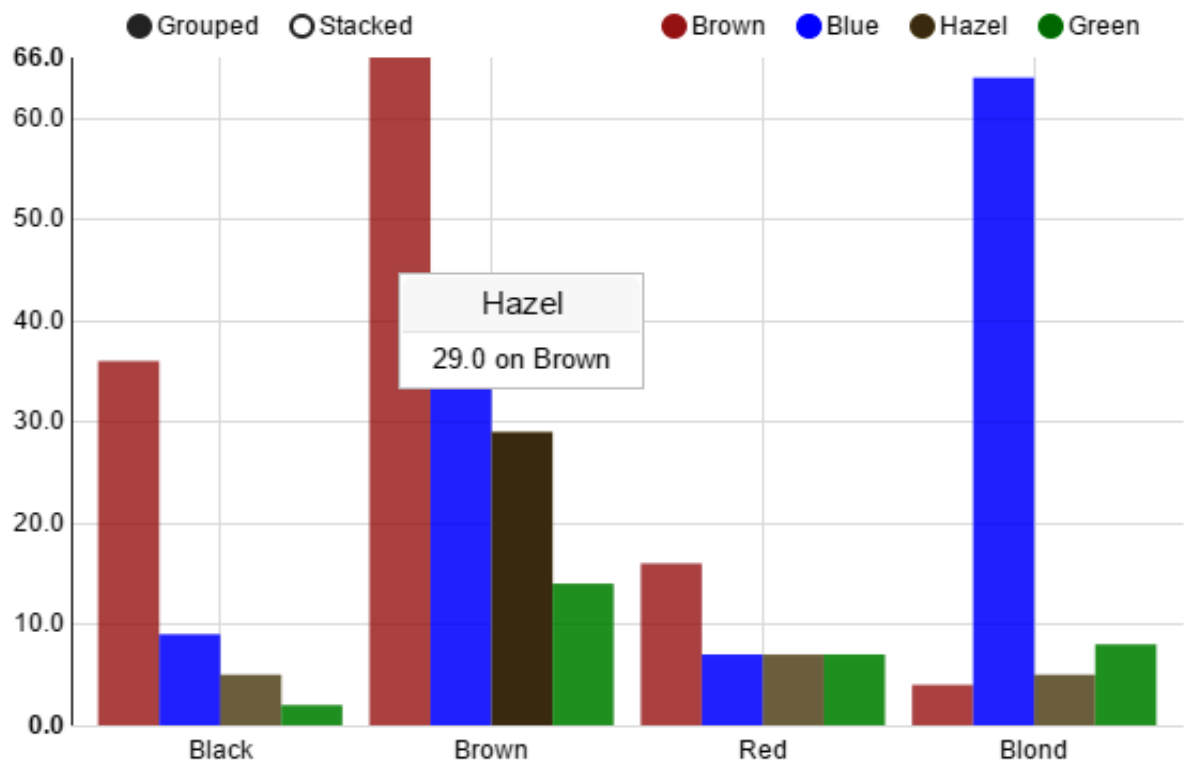


Figura 4.16. Ejemplo del tipo de gráfica *NVD3* de *rCharts*.

A continuación, se describirá el proceso de implementación de estas gráficas para cada archivo de *Shiny:*

ui.R

La pestaña de las gráficas de barras es bastante parecida a las lineales.

Primero, se establece un `sidebarLayout` que sitúa la barra de control fija en la izquierda y la gráfica en la derecha.

El **panel de control** tiene la siguiente estructura:

- Primero, tres `radioButtons` que permiten elegir el tipo de variable a medir (tipo de incidente, valoración y tipo de recurso).

- Debajo, otros tres `radioButtons` para elegir la variable temporal (Año, mes o hora). Al seleccionar esta variable, con un `conditionalPanel` se despliega una lista desplegable para elegir el valor de la variable temporal deseado ("2010", "Enero", "Marzo", "14:00", etc.).

server.R

En este fichero se describirá el procesamiento y filtrado de datos que se hace en tiempo de ejecución para las gráficas de barras.

El primer paso es recoger los valores de las variables seleccionadas por el usuario e intercambiarlas por códigos para facilitar el procesamiento.

Posteriormente, se implementan los condicionales que cubren todas las combinaciones de variables posibles. Una vez dentro de una condición, se filtra el subconjunto de datos correspondiente para esa combinación de variables (creado en los capítulos anteriores), se calcula el porcentaje que representa el número de incidentes de cada agregación respecto al total.

A continuación, nos quedamos con los 6 mayores valores. El subconjunto final tendrá un aspecto parecido al de la Figura 4.17:

	ISLA	TIPO_RECURSO	perc
1	Fuerteventura	Bomberos	0.137
2	Gran Canaria	Bomberos	1.550
3	La Palma	Bomberos	0.130
4	Lanzarote	Bomberos	0.153
5	MAR	Bomberos	0.001
6	Tenerife	Bomberos	1.797
7	Lanzarote	Otros	0.165
8	La Palma	Otros	0.187
9	Tenerife	Otros	2.043
10	Fuerteventura	Otros	0.156
11	MAR	Otros	0.001
12	La Gomera	Otros	0.057
13	Gran Canaria	Otros	1.105
14	La Palma	Guardia Civil	0.467
15	Lanzarote	Guardia Civil	0.494

wing 1 to 15 of 38 entries

Figura 4.17. Subconjunto de datos para gráfica de barras.

Cabe destacar varias características de las propias gráficas interactivas:

- Sobre la gráfica se encuentra la leyenda de colores y valores de la variable. Clicando sobre cada elemento se habilitan o deshabilitan las barras de ese valor.
- Cuando se deshabilita una barra, los rangos del eje Y se redimensionan para adaptarse a los valores numéricos de las barras restantes.
- Además, existen dos botones superiores "*Stacked*" y "*Grouped*", que apilan o agrupan las barras a voluntad del usuario.

A continuación en la Figura 4.18 se muestra la interfaz acabada.

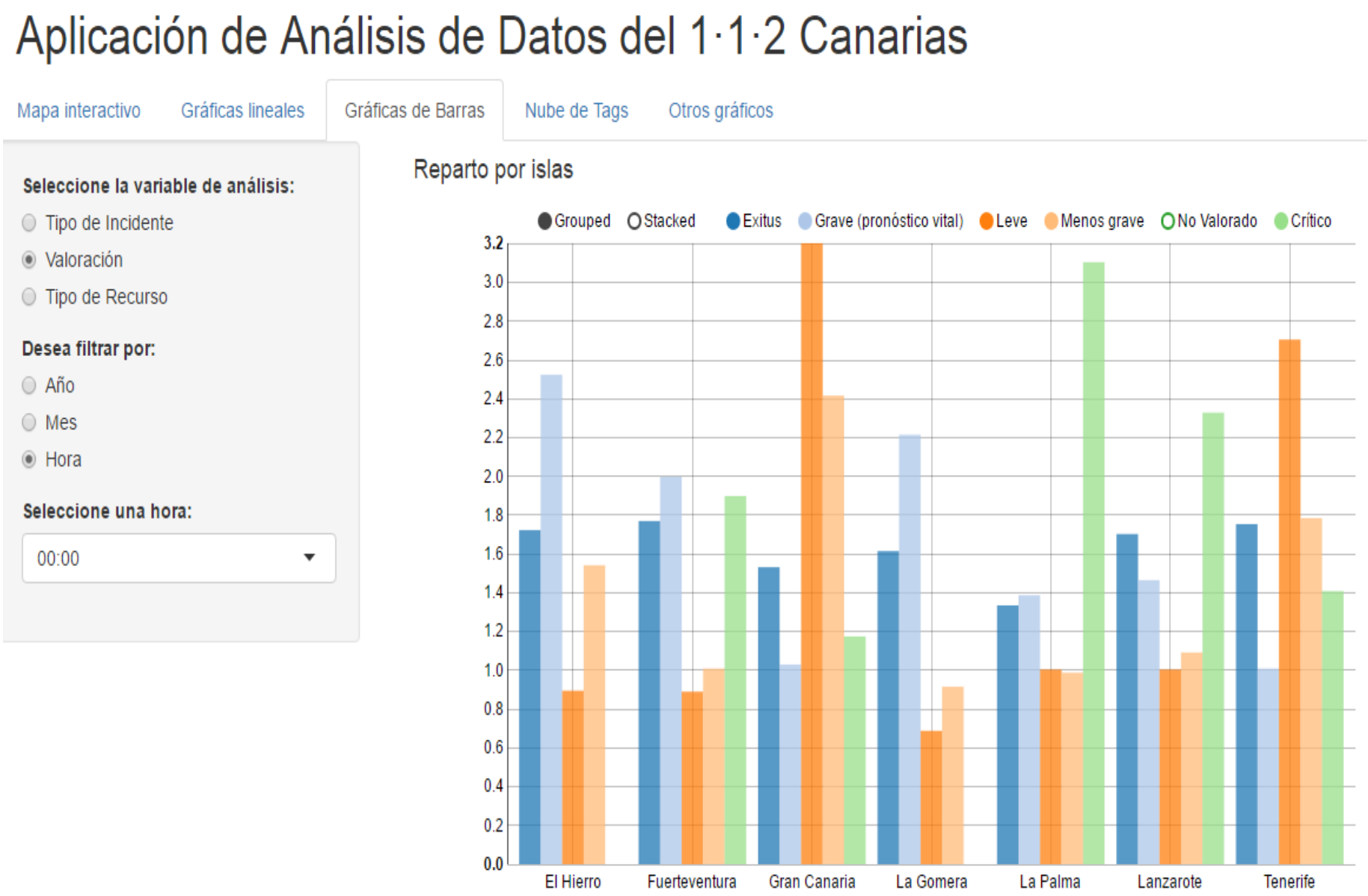


Figura 4.18. Gráfica de barras interactiva.

4.6.5 Nube de *tags*

Para la creación de la nube de palabras se ha usado un paquete de *R,* llamado *wordCloud*, que permite crear nubes de palabras usando una matriz término-nº ocurrencias.

A continuación, se describirá el proceso de implementación de estas gráficas para cada archivo de *Shiny:*

ui.R

Primero, se establece un `sidebarLayout` que sitúa la barra de control fija en la izquierda y la nube en la derecha.

El **panel de control** tiene la siguiente estructura:

- Un *slider* que controla el número de ocurrencias que tiene que tener un término para aparece en la nube de tags.
- Otro *slider* que controla el número máximo de palabras que puede albergar la nube.

server.R

En este fichero se describirá el procesamiento y filtrado del texto que se hace en tiempo de ejecución para la nube de términos.

El primer paso es una llamada a la función `getTermMatrix` (su funcionamiento se explicará posteriormente), la cual devuelve una matriz de términos-nº ocurrencias.

A continuación se llama a la función `wordcloud_rep`, que lleva como parámetros la matriz antes hallada, los valores de los *sliders*, la paleta de colores para las palabras, así como el tamaño de la nube.

global.R

En este fichero se almacena el código de la función para que esté precargada al ejecutar la aplicación y se pueda llamar en cualquier momento.

La función `getTermMatrix` es descrita detalladamente a continuación:

- Se usa un `Corpus` de la librería *tm* de *R* para cargar el el texto que contiene todas las descripciones de la variable **DESCRIP** concatenadas.

Este elemento se usa en minería de datos para almacenar grandes cantidades de texto y operar sobre él.

- Primero, se realizan varias operaciones sobre él: poner todas las palabras en minúscula, eliminar signos de puntuación y números, y, finalmente, eliminar las palabras sin valor (preposiciones, artículos, etc.).
- Posteriormente, se crea la matriz de términos a partir de ese corpus usando la función `TermDocumentMatrix`. Finalmente, se ordena la matriz por ocurrencias.

En la Figura 4.19 se puede ver el aspecto final de la pestaña.

Figura 4.19. Nube de *tags*.

Capítulo 5.
Conclusiones y líneas futuras

La Ciencia de los Datos y el *Big Data* son el futuro. Puede que tal afirmación suene pretenciosa sobre todo teniendo en cuenta la infinidad de tecnologías que se desarrollan en la actualidad, pero cada vez más el reto que supone el análisis de unos datos que aumentan en volumen de manera exponencial y los beneficios derivados del correcto tratamiento de los mismos, hace que ambas áreas de forma combinada sean a día de hoy de las tecnologías con mayor proyección a nivel mundial.

La situación en el marco español es ligeramente diferente, porque si bien en Estados Unidos, Alemania, o Francia la tecnología del análisis de grandes cantidades de datos se encuentra en pleno auge, en España se están dando los primeros pasos en ese sentido de la mano de grandes empresas como BBVA. Esto supone un desfase de alrededor de dos años con respecto a los países punteros en estos ámbitos por lo que representa una oportunidad de presente en el extranjero o a corto plazo a nivel nacional.

Al ser un proyecto multidisciplinar, las conclusiones obtenidas son diversas, más incluso cuando no se ha limitado el alcance del proyecto a un desarrollo meramente académico, sino que se le ha otorgado de un valor extra al realizarse de manera colaborativa con una entidad externa tal como el CECOES 1-1-2 de Canarias.

Siguiendo esta última idea, la primera conclusión que se obtiene es aumento del valor del proyecto que se ha conseguido al darle un enfoque eminentemente práctico, con datos reales y elaborando una solución útil y exportable a este problema.

El hecho que de los datos de entrada sean reales añade un componente extra como es la variabilidad de los mismos, pues los registros analizados recogen de igual manera errores en la recopilación de información, aportan información adicional sobre momentos importantes como incendios o temporales, etc.

Esto no hace más que enriquecer la solución puesto que obliga a generar un código mucho más compacto que integre una cierta tolerancia a esos errores.

De forma paralela, sirve como puente entre el ámbito académico y el mundo empresarial (o en este caso, de los organismos públicos) convirtiéndose en una primera piedra de toque a la hora de realizar proyectos con terceros, dependiendo en muchos momentos de ellos para el correcto devenir del mismo.

A pesar de que en ocasiones ha podido resultar frustrante el tiempo de espera inicial (para la obtención de datos y acuerdos de confidencialidad) debido a la carga de trabajo propia de una entidad como CECOES, que, en un principio, hubo retrasado el avance del proyecto, el contar con gente tan preparada y con tanta experiencia lo compensa con creces.

El desarrollo del proyecto no ha hecho más que afianzar mi idea inicial de que el mundo de la Ciencia de los Datos presenta un futuro muy prometedor y que se encuentra en un momento excepcional para adentrarse en el mismo, con un lenguaje y un entorno que ya lo siento como propio, como es *R*, y la aparición de nuevos *frameworks* de desarrollo para combinarlo con *Big Data*, como puede ser principalmente *Spark*.

Si bien esto no quita la dificultad del uso de estas tecnologías, la curva de aprendizaje no es corta, debido a la gran cantidad de posibilidades. Además de que exige un conocimiento tanto informático, como estadístico si de verdad se quiere dar un salto cualitativo en este área.

Por otra parte, este proyecto no pretende pararse aquí, sino desde el CECOES existe la intención de continuar con el mismo, tratando de implantar esta herramienta en su sistema con la idea futura de que funciona con un flujo de datos en tiempo real. A pesar de ser una intención ambiciosa, no cabe duda que más de la mitad de camino ya está recorrido con la realización de este proyecto.

Summary and Conclusions

Data Science and Big Data are the future. This might sound like a pretentious statement especially bearing in mind the countless number of technologies currently being developed. However, the challenge posed by the analysis of data, increasing in volume at an exponential rate, and the benefits stemming from the correct processing of such data, makes both work areas combined to be currently among the technologies with a greater impact on a global level.

The situation in Spain however is slightly different. While in the USA, Germany or France, the technology of analyzing large amount of data is now-a-days in a boom, in Spain, technology is making its first steps in that direction being led by large companies, such as BBVA bank. This suggests a lag of about two years with respect to the leading countries in these two fields, and indicates an opportunity abroad or in the short term on a national level.

Being a multi-disciplinary project, the conclusions obtained are diverse, even more when the project scope has not been limited to a merely academic undertaking, but rather it has gained added value by being undertaken collaboratively with an external partner, such as the CECOES 1-1-2.

Continuing with this last idea, the first conclusion that was reached is the increase in the value of the project that has been achieved, thanks to the completely practical character, with real data and creating a useful and exportable solution to the matter under consideration.

The fact that the input data was real, really adds an extra component like the variability of such data, as the analyzed records also contained errors during the information gathering by the operators, give additional information about key incidents like forest fires or storms, etc.

This only enriches the solution, inasmuch it forces the generation of code which is much more compact and allows a certain tolerance to errors.

In parallel, is serves as a bridge between the academic environment and the business world (or in this specific case, public institutions world), converting

itself into the initial touchstone when it comes to carrying out the project with third parties, often depending on them for the correct outcome of it. Despite the fact that occasionally it could have been frustrating that the initial waiting time (for data collection and confidentiality agreements), due to the workload for an institution such as the CECOES has initially hold the project back, having people that prepared and with that much experience around more than offsets it.

The development of the project has only strengthened my initial idea that the world of Data Science presents a promising future and it is an exceptional moment to embark on it, with a solid programming language and environment such as R, and the emergence of new development frameworks to combine it with Big Data technologies, such as Spark.

Although it is still difficult to learn how to use these technologies, the learning-curve is not short, due to the great amount of possibilities. Besides this takes detailed knowledge of both Computer Science and Statistics if you really want to take a step forward in this area.

On the other hand, this project is not stopping at this point, since the CECOES has the intention of continuing to work on it and they expect to implement the application in their system with the idea that the tool will work with a real-time data flow. In spite of being an ambitious intention, there is no doubt that we are already halfway from this goal with this project.

Capítulo 6. Presupuesto

En este proyecto, la cantidad de hardware necesario depende de las necesidades y aspiraciones de rendimiento del CECOES. Para realizar estas estimaciones usaremos unos parámetros estándar que nos permitan establecer una aproximación fiable del coste total que llevaría realizar este proyecto y ponerlo en funcionamiento de forma estable.

6.1 Recursos hardware

Partiendo de que el CECOES no posea ningún dispositivo disponible donde desplegar la aplicación, el coste aumentará al ser necesario adquirir equipamiento que sea capaz de dar cobertura a la demanda actual y futura en términos de almacenamiento y procesamiento de información. Considerando que se va a implantar el sistema con fines profesionales y no académicos, podemos encontrarnos con una inversión de 7.000 € en un servidor HP ProLiant DL380p Gen8 (diseñado para procesos con grandes cantidades de datos) hasta soluciones mucho más económicos como un servidor HP ProLiant ML350 Gen9 por 2.000 €.

6.2 Recursos software y licencias

Una de las grandes ventajas de este proyecto fue la idea de utilizar únicamente software libre que permitiera un gasto nulo en licencias. En nuestro caso particular R cumple estos requisitos, además de poseer una enorme comunidad de usuarios que contribuyen con mejoras continuas y nuevas características.

6.3 Recursos humanos

La instalación y configuración, así como el desarrollo del proyecto, sí que supone una gran inversión de tiempo. Para este caso, al tratarse de un proyecto individual estimamos la necesidad de un único Ingeniero Informático. Si establecemos un sueldo estándar de 20 €/h, durante las 300 horas que oficialmente ocupa el Trabajo de Fin de Grado (tiempo empleado para acabar el proyecto), obtenemos unos requisitos de 6.000 €.

6.4 Coste total

Considerando todos los apartados anteriores obtenemos un presupuesto mínimo de 8.000 €. Teniendo en cuenta el valor del proyecto, así como los beneficios que reporta al cliente en términos de conocimiento interno e información valiosa, además del coste de cualquier proyecto de análisis y procesamiento de grandes cantidades de datos, se llega a la conclusión de que sería una inversión fácilmente asumible.

Bibliografía

[1] Hadley Wickham. *R for Data Science*. - r4ds.had.co.nz

[2] R Project - r-project.org

[3] R Studio - rstudio.com

[4] R Shiny - shiny.rstudio.com

[5] ISTAC - www.gobiernodecanarias.org/istac

[6] El Científico de Datos: una novedosa profesión - noticias.universia.es/ciencia-nn-tt/noticia/2014/05/06/1095994/cientifico-datos-novedosa-necesaria-profesion.html

[7] Ramnath's Github - ramnathv.github.io

[8] Highcharts - highcharts.com

[9] HighchartsUtils Github - github.com/bthieurmel/highchartsUtils

[10] Data Science L.A. Youtube Channel – youtube.com/channel/UCJ4D3tL-iEWyJJDXJAsd9Mw

[11] Blackspot Github - github.com/blmoore/blackspot

[12] Stack Overflow - stackoverflow.com

[13] Big Data Applications and Analytics Course - bigdatacourse.appspot.com

[14] Wikibon - Executive Summary: Big Data Vendor Revenue and Market Forecast, 2011-2016 wikibon.com/executive-summary-big-data-vendor-revenue-and-market-forecast-2011-2026

[15] Estudio de The Economist – www.capgemini.com/news/capgemini-report-shows-rising-impact-of-big-data-on-decision-making