Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Estudio del turismo en Tenerife y desarrollo de una aplicación en Android

Study of tourism in Tenerife and developing an application for Android

Alberto Delgado Soler

La Laguna, 07 de septiembre de 2021

D. **Alejandro Pérez Nava**, con N.I.F. 43.821.179-S profesor Asociado de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como tutor

D. **Fernando Pérez Nava**, con N.I.F. 42.091.420-V profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como cotutor

C E R T I F I C A N

Que la presente memoria titulada:

"Estudio del turismo en Tenerife y desarrollo de una aplicación en Android"

ha sido realizada bajo su dirección por D. **Alberto Delgado Soler**, con

N.I.F. 42.224.645-A.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 07 de septiembre de 2021.

Agradecimientos

Este trabajo de fin de grado ha sido posible gracias a la tutorización de D. Alejandro Pérez Nava, el cual me guió en la realización del mismo, aconsejándome y enfocando el proyecto en el buen camino.

También agradecer a mis compañeros de carrera Sergio González Guerra y Tomás González Martín, con los cuales he aprendido y disfrutado de la vida universitaria.

Fuera del ámbito académico, agradecer a mi familia y amigos de la infancia los cuales también han estado apoyándome en todo momento desde que tengo memoria.

Por último agradecer a mi pareja Ana Garrido Benito que me hace ser la mejor versión de mí mismo, aunque ella diga lo contrario.

Licencia

Resumen

El objetivo de este trabajo ha sido el diseño de una aplicación para Android, para ello la idea que se propuso fue la de una aplicación que mostrara los datos que se pueden obtener en la página web TenerifeData[1], concretamente los del apartado "Turismo".

Al observar datos del turismo a lo largo del tiempo podemos ver la evolución del sector en la isla de Tenerife, incluyendo el efecto que ha tenido el COVID-19, llegando a bajar la cantidad de turistas gravemente a cifras que no se veían desde hace 30 años, según los la página web de la que obtendremos los datos.

El proceso fue el siguiente, se almacenaron los datos en una base de datos propia para poder manipular los datos de origen libremente, se hizo que los datos de la base de datos se actualicen periódicamente, luego se desarrolló una aplicación que consulta la base de datos en la que el usuario decide que datos observar, mostrándose una gráfica de éstos acompañada de información útil, siendo éstos la media, la desviación típica, la mediana y la moda de los datos.

Palabras clave: Android, Turismo, Tenerife, Base de datos, Aplicación.

Abstract

The main objective of this project is designing an application for Android, the idea is to show data from the website TenerifeData[1], which gathers data about the island of Tenerife. In particular the data will come from the section "Tourism".

When we look at tourism in Tenerife over time, we can see the development of the tourism industry on the island, including the damaging impact of COVID-19, taking the quantity of tourists down to fatal levels that we have not seen for 30 years, according to the webpage that we are gathering the data from.

The steps were the following, store the data in our own database so we can freely manipulate it, make that data update periodically, then develop an application that retrieves the data from our database and the user selects which data he wants to see. Said data is shown in a chart along some useful information, that information is the mean, the standard deviation, the median and the mode of the data.

Keywords: Android, Tourism, Tenerife, Database, Application.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1 Introducción

1.1 Motivo

"Canarias es, tras Cataluña, la segunda comunidad autónoma preferida entre los turistas internacionales en España. Durante décadas, este archipiélago de origen volcánico situado frente a las costas de Marruecos y el Sáhara Occidental ha sido casi un lugar de peregrinación para turistas de toda Europa, incluido el resto de España. Su clima subtropical y su riqueza cultural y paisajística hacen de las "islas Afortunadas" un destino turístico perfecto, algo que la población local ha aprendido a explotar muy bien. Tal es la importancia del turismo para la economía regional, que el sector ha llegado a generar más del 40% del empleo y el 35% del PIB canario." [2]

El trabajo de fin de grado que se va a desarrollar consiste en el estudio del turismo de Tenerife y la realización de una aplicación para Android. Se usarán datos actualizados hasta 2021 por lo que se podrá observar el impacto que ha supuesto el Covid-19 en el ámbito del turismo.

1.2 Fuente de la información

La información proviene de la página web "www.tenerifedata.com", en el apartado turismo filtramos en las etiquetas por turismo y obtenemos 7 conjuntos de datos: Turistas alojados por categoría y zona, Estancias medias por nacionalidad y tipo de establecimiento, Pernoctaciones, Índices de ocupación, Oferta de alojamiento, Turistas alojados por municipios y Actividad turística. Éstos son los conjuntos de datos que usaremos, y cada uno puede tener uno o varios archivos asociados.

1.3 Pasos a seguir

1. El primer paso consiste en el desarrollo de una base de datos que se actualice cuando los datos de la página web TenerifeData sean actualizados. Esta base de datos es la que luego consultaremos con la aplicación de Android.

2. Acto seguido se desarrollará la aplicación de Android. Primero se pedirá qué conjunto de datos se quiere consultar, seguidamente que tabla del conjunto de datos y por lo tanto consultamos la base de datos que hemos creado para poder mostrar gráficos y datos interesantes de la tabla según la variable que se quiera visualizar.

1.4 Estructura del documento

- **Capítulo 2 Entorno de trabajo**

  Descripción de las herramientas utilizadas para el desarrollo del trabajo de fin de grado.

- **Capítulo 2 Estudio de los datos**

  Entenderemos qué datos se recogen y como interpretarlos para su correcta visualización en la aplicación para Android.

- **Capítulo 3 Desarrollo de la base de datos**

  Pasos seguidos para realizar la base de datos, incluyendo un script que actualiza los datos con respecto a la página de origen de los datos.

- **Capítulo 4 Desarrollo de la aplicación para Android**

  Pasos seguidos para el desarrollo de la aplicación, para ello realizamos la consulta a la base de datos, interfaz y código de la aplicación.

- **Capítulo 5 Conclusión y líneas futuras**

  Reflexión final sobre los resultados y posibles mejoras futuras.

- **Capítulo 6 Summary and conclusions**

  Versión en inglés del capítulo anterior.

- **Capítulo 7 Presupuesto**

  Costes de los diferentes apartados, teniendo en cuenta la horas de planificación y las de desarrollo de cada uno.

Capítulo 2 Entorno de trabajo

2.1 Sistema operativo

El sistema operativo Ubuntu fue el utilizado durante el desarrollo, más concretamente la versión 18.04.

Como podemos ver en [3], Ubuntu es una distribución de Linux de software libre basada en Debian. Principalmente es usada para ordenadores personales pero también puede usarse a modo de servidor. Utiliza una interfaz gráfica de usuario y aplicaciones de escritorio de Linux similar a la interfaz de escritorio de Windows.

Es el sistema en el que se han instalado todas las herramientas utilizadas, como son la base de datos MySQL y la herramienta de desarrollo para Android llamada Android Studio.

2.2 Base de datos

Para el desarrollo de la base de datos se ha utilizado MySQL, versión 14.14. Como se define en [4], "MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional."

También podemos ver en [4] que MySQL utiliza un lenguaje de consulta estructurado o Structured Query Language (SQL) para el manejo de las bases de datos. Las declaraciones SQL indican al servidor acciones de creación, consulta y manipulación de los datos, además de poder controlar quien puede realizar esas acciones.

Además "MySQL no es el único (R)DBMS que hay en el mercado, pero es uno de los más populares y solo está por detrás de Oracle Database cuando se califica utilizando parámetros críticos como la cantidad de menciones en los resultados de búsqueda, los perfiles profesionales en LinkedIn y la frecuencia de discusiones técnicas en foros de internet. El hecho de que muchos de los principales gigantes de la tecnología confíen en él refuerza aún más esta merecida posición." [4].

2.3 Editor de texto

Durante el desarrollo de la base de datos también se desarrolló un script en el lenguaje PHP y para ello se utilizó el editor de texto Kate. como muchos editores de texto, dispone de resaltado de código para facilitar la lectura y escritura, autorelleno para agilizar la escritura, soporta una gran cantidad de lenguajes de programación (entre ellos PHP) y además de la posibilidad de extender sus funcionalidades mediante scripts.

Por otra parte, para la realización de este documento usaremos LibreOffice Writer, que es un programa de software libre para la edición de documentos similar al famoso Microsoft Word.

2.4 Entorno de desarrollo integrado (IDE)

Para el desarrollo de la aplicación se utilizó la herramienta Android Studio. Se pueden desarrollar las aplicaciones en Java o en Kotlin (nombrado por Google como lenguaje oficial de Android, al mismo nivel que java[5]).

El editor de texto integrado en Android Studio tienes las funciones esperadas de un editor de texto de un entorno de desarrollo, como pueden ser autocompletado de palabras y bloques de código enteros, resaltado de texto, renombrado de elementos del código a través de varios archivos, indentación automática.

También facilita la utilización de librerías de código externas que pueden ser buscadas directamente en Android Studio a través de internet y otra de las capacidades disponibles es la de enlazar el proyecto a un repositorio git para poder tener un control de versiones integrado.

Por último cuando para comprobar el funcionamiento del programa se puede conectar un dispositivo Android para que ejecute el programa o en el propio AndroidStudio puedes crear una máquina virtual que simule un dispositivo Android.

2.5 Control de versiones

El control de versiones es una herramienta de gran utilidad y el más popular es Git [6], el cual es un sistema de control de versiones distribuido de software libre. Contaremos con un repositorio local y uno remoto que alojaremos en la página web GitHub, la cual añade funcionalidades como un gestor de proyectos que ayuda en la organización.

2.7 Programador de tareas

Para la realización de este proyecto, usamos un planificador de tareas llamado Anacron. Se usa en este proyecto para que ejecute periódicamente un script PHP que actualiza la base de datos. Una de las ventajas de Anacron es que si cuando tiene que ejecutarse una tarea programada no se encuentra en funcionamiento el sistema, la tarea será ejecutada cuando vuelva a estar en funcionamiento. En esto se diferencia de Cron que es otra herramienta de programación de tareas.

2.8 Editor gráfico

Como es una aplicación para Android, necesitamos un logo que se muestre en el Escritorio del sistema Android, para ello usaremos la herramienta Gimp, versión 2.10.24. Además también vamos a desarrollar una portada que se muestre cuando se abra la aplicación.

Según sus desarrolladores [7], es una herramienta de distribución gratuita con la que realizar tareas como retoque de fotografías, composición de imágenes, pintar de forma similar a la famosa herramienta Paint, animación y muchas cosas más. Además se le pueden añadir extensiones.

Capítulo 3 Estudio de los datos

3.1 Descripción de los conjuntos de datos

- **Índice de ocupación por plazas**: se define como la relación expresada en porcentaje entre el total de las pernoctaciones en un periodo de tiempo determinado y el producto de las plazas, por el número de días que tiene ese período.

- **Pernoctaciones**: cantidad de noches que se han alojado los turistas.

- **Turistas alojados por categoría y zona**: pernoctaciones diferenciando de qué país provienen los turistas.

- **Oferta de alojamiento**: plazas de alojamiento turístico en Tenerife según categoría y zona, datos semestrales.

- **Turistas alojados por municipios**: turistas alojados en establecimientos turísticos en los municipios de Adeje, Arona, Puerto de la Cruz y Santa Cruz de Tenerife.

- **Estancias medias por nacionalidad y tipo de establecimiento**: las estancias medias por nacionalidad, se definen como la división entre el total de las pernoctaciones y el número de turistas alojados, por nacionalidad en el periodo de referencia (serie anual).

- **Actividad turística**: información sobre los alojamientos hoteleros, extrahoteleros y agencias de viajes.

3.2 Descripción de las tablas

A continuación se explicarán diversas características de las tablas para entender los datos que se proporcionan.

Las tablas con las que se está trabajando tienen en cada fila el campo año y el campo mes (o semestre en algún caso). Gracias a esto tendremos los valores de cada mes de cada año, pero además en el campo mes se incluye el mes 13 para englobar todo el año.

En la descripción de los conjuntos de datos anterior, se nombraron los términos “por categoría” y “por zona”. La definición de éstos términos es la siguiente:

- **Categoría**: Las tablas que se muestran por categoría muestran los datos separando por estrellas, es decir, separa por 5, 4, 3, 2 y 1 estrellas. También muestran datos separando hoteles y establecimientos extrahoteleros. Los establecimientos extrahoteleros son todos aquellos que no sean hoteles, como alojamientos rurales, campings, viviendas particulares, apartamentos turísticos, etc. También se muestra el valor total de los datos.

año	mes	nacion	1 estrella	2 estrellas	3 estrellas
2010	1	España	898	4365	16498
2010	1	Holanda	80	164	1434
2010	1	Bélgica	58	97	969
2010	1	Alemania	402	656	7349

Figura 1: Ejemplo de los datos, representa los turistas por categorías. Parte 1.

4 estrellas	5 estrellas	hoteleros	extrahoteleros	total
33170	6804	61735	14977	76712
3429	491	5598	4945	10543
6373	1564	9061	2324	11385
26814	5127	40348	11110	51458

Figura 2: Ejemplo de los datos, representa los turistas por categorías. Parte 2.

- **Zona**: Las tablas que se muestran por zona diferencian 4 zonas en las isla de Tenerife. La zona 1 es Santa Cruz de Tenerife, la zona 2 es La laguna, Tegueste y Tacoronte, la zona 3 es Puerto de la Cruz y el resto del Norte de la isla y la zona 4 sería el resto de la isla como por ejemplo Adeje, Arona, Candelaria y demás. Teniendo esto en cuenta se suele mostrar el dato de los hoteles de la zona, los establecimientos extrahoteleros de la zona y el valor total de la zona. También se muestra el valor total de los hoteles y extrahoteleros en toda la isla y el valor total completo.

año	mes	hoteles zona 1	extrahoteleros zona 1	total zona 1	hoteles zona 2
1978	1	45,91	81,91	49,77	58,34
1978	2	48,94	83,65	52,66	58,82
1978	3	49,95	81,03	53,28	56,96
1978	4	42,95	85,46	47,5	39,96
1978	5	33,65	85,12	39,16	20,47
1978	6	38,83	84,34	43,7	17,8
1978	7	47,11	85,46	51,22	27,78
1978	8	50,37	85,24	54,1	44,92

Figura 3: Ejemplo de los datos, representa el índice de ocupación por zonas. Parte 1.

extrahoteleros zona 2	total zona 2	hoteles zona 3	extrahoteleros zona 3
54,74	56,79	81,68	49,05
46,77	53,64	86,62	48,73
53,79	55,6	77,67	48,77
18,23	30,62	65	31,79
11,54	16,63	59,08	28,57
7,14	13,22	56,57	38,63
34,28	30,58	75,59	55,34
58,04	50,56	85,45	63,79

Figura 4: Ejemplo de los datos, representa el índice de ocupación por zonas. Parte 2.

total zona 3	hoteles zona 4	extrahoteleros zona 4	total zona 4
71,15	71,37	63,03	66,1
74,39	66,08	61,05	62,9
68,34	69,45	55,94	60,91
54,28	51,66	38,96	43,63
49,23	43,19	26,15	32,42
50,78	42,51	21,68	29,34
69,05	43,16	51,11	48,18
78,46	67,33	57,08	60,85

Figura 5: Ejemplo de los datos, representa el índice de ocupación por zonas. Parte 3.

hoteles total	extrahoteleros total	total
75,11	57,6	67,45
77,08	56,03	67,86
72,4	53,44	64,1
58,67	35,88	48,69
51,19	27,13	40,65
49,76	28,2	40,31
62,8	52,37	58,23
76,14	60	69,07

Figura 6: Ejemplo de los datos, representa el índice de ocupación por zonas. Parte 4.

Capítulo 4 Desarrollo de la base de datos

4.1 Planteamiento de la base de datos

Para el desarrollo de la base de datos se ha planteado que se tenga que actualizar cada cierto tiempo. Para ello se ha realzado un script en lenguaje PHP que, si se han actualizado los datos en tenerifedata.com, actualiza las tablas de la base de datos.

Primero se definieron las tablas en la base de datos, ya que leyendo los archivos CSV en los que se encuentran los datos no se definen los tipos de las columnas y en la base de datos es necesario definir el tipo de variable.

Field	Type	Null	Key	Default	Extra
año	int(11)	NO	PRI	NULL	
mes	int(11)	NO	PRI	NULL	
nacion	varchar(255)	NO	PRI	NULL	
1 estrella	int(11)	YES		NULL	
2 estrellas	int(11)	YES		NULL	
3 estrellas	int(11)	YES		NULL	
4 estrellas	int(11)	YES		NULL	
5 estrellas	int(11)	YES		NULL	
hoteleros	int(11)	YES		NULL	
extrahoteleros	int(11)	YES		NULL	
total	int(11)	YES		NULL	

Figura 7: Ejemplo de la definición de una tabla. La tabla mostrada es Turistas por categoría 2010 - 2021.

El segundo paso fue definir una tabla, llamada “tablesInfo” que recoja información de cada tabla, los datos que recoge son los siguientes:

- Nombre del archivo .csv del cual obtenemos los datos. Este también es el nombre con el que definimos las propias tablas en nuestra base de datos ya que no contiene tildes ni espacios, por los posibles problemas que pueda causar.
- Fecha de la última actualización para comprobar si existe una versión más reciente en TenerifeData.
- Nombre de los datos, el cual contiene tildes y espacios para usarse desde la aplicación Android.
- Conjunto de datos al que pertenece, así cuando consultemos los datos en la aplicación tendremos las tablas de forma ordenada.

Este es el resultado:

table_name_First20Chars	last_update	displ_name_First20Chars	data_set_First20Chars
estanciasmediasporna	2021-05-28 09:53:41	Estancias medias por	Estancias medias por
indicesdeocupacionpo	2021-09-03 12:50:05	Índices de ocupación	Índices de ocupación
indicedeocupacionpor	2021-09-03 12:50:48	Índices de ocupación	Índices de ocupación
plazasporcategorias	2021-07-29 12:10:31	Plazas por categoría	Oferta de alojamient
plazas-por-zonas	2021-07-29 12:11:23	Plazas por zonas	Oferta de alojamient
pernoctacionesporcat	2021-09-03 12:54:42	Pernoctaciones por c	Pernoctaciones
pernoctacionesporzon	2021-09-03 12:55:14	Pernoctaciones por z	Pernoctaciones
turistaporcategorias	2021-09-03 12:52:08	Turistas por categor	Turistas alojados po
turistasporcategoria	2019-06-25 09:09:43	Turistas por categor	Turistas alojados po
turistasporcategoria	2019-06-25 09:12:49	Turistas por categor	Turistas alojados po
turistasporcategoria	2019-06-25 09:18:06	Turistas por categor	Turistas alojados po
turistasporzonas	2021-09-03 12:53:03	Turistas por zonas 2	Turistas alojados po
turistasporzonas2000	2019-06-25 09:27:08	Turistas por zonas 2	Turistas alojados po
turistasporzonas1990	2019-06-25 09:29:51	Turistas por zonas 1	Turistas alojados po
turistasporzona	2019-05-22 12:25:58	Turistas por zonas 1	Turistas alojados po
turistasporcategoria	2019-05-22 12:26:23	Turistas por categor	Turistas alojados po
turistasalojadosadej	2020-04-22 12:10:20	Turistas alojados en	Turistas alojados po
turistasalojadosaron	2020-04-22 12:12:27	Turistas alojados en	Turistas alojados po
turistasalojadospuer	2020-04-22 12:13:59	Turistas alojados en	Turistas alojados po
turistasalojadossant	2020-04-22 12:15:19	Turistas alojados en	Turistas alojados po

Figura 8: Tabla que recoge la información necesaria de las demás tablas. Se muestran los primeros 20 caracteres de cada columna.

Al final tendremos 21 tablas, contando la tabla que recoge información de las demás tablas.

TABLE_NAME
estanciasmediaspornacionalidadtipodeestablecimiento
indicedeocupacionporzonas
indicesdeocupacionporcategorias
pernoctacionesporcategorias
pernoctacionesporzonas
plazas-por-zonas
plazasporcategorias
tablesInfo
turistaporcategorias
turistasalojadosadeje
turistasalojadosarona
turistasalojadospuertodelacruz
turistasalojadossantacruz
turistasporcategoria
turistasporcategorias19781989
turistasporcategorias19901999
turistasporcategorias20002009
turistasporzona
turistasporzonas
turistasporzonas19901999
turistasporzonas20002009

Figura 9: Listado de tablas en nuestra base de datos.

4.2 Planteamiento del script PHP

A partir de aquí se desarrollaría el script en PHP para actualizar la base de datos, ya que al principio está vacía. Se ha decidido utilizar este lenguaje ya que estoy familiarizado con él y está estrechamente relacionado con las bases de datos.

Para poder realizar esto, en resumen, si la fecha de actualización para un archivo CSV en tenerifedata.com es posterior a la que está registrada para esa tabla en nuestra base de datos, se realiza una operación TRUNC que elimina todos los datos de la tabla (por si se ha actualizado también información antigua) y se recorre el archivo CSV construyendo las consultas INSERT y ejecutándolas.

Para que se ejecute continuamente se usa la herramienta "Anacron" de Linux. Gracias a esta herramienta podemos planificar que un comando se ejecute cada cierto tiempo. Se eligió que se ejecutara diariamente, ya que observando las fechas de actualización de los archivos se puede comprobar que no se actualizan todos a la vez, y si el ordenador no está encendido en el momento que debe ejecutarse la tarea, se ejecutará la próxima vez que se encienda el ordenador.

```
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
HOME=/root
LOGNAME=root

# These replace cron's entries
1   5   tenerifedataTurismoUpdate /usr/bin/tenerifedataTurismoUpdate
1   5   cron.daily  run-parts --report /etc/cron.daily
7   10  cron.weekly run-parts --report /etc/cron.weekly
@monthly    15  cron.monthly    run-parts --report /etc/cron.monthly
```

Figura 10: Configuración del fichero /etc/anacrontab

4.3 Descripción del script

Nos conectamos a la base de datos con la función "mysqli_connect()", simplemente tenemos que proporcionar la IP en la que se encuentra la base de datos, el usuario con el que nos autenticaremos y su contraseña.

Para obtener los datos haremos uso de la api de la página web. Obtendremos un objeto tipo JSON por cada conjunto de datos y este objeto JSON contiene un array de objetos JSON con cada archivo del conjunto de datos. Podemos comprobar los atributos para obtener el tipo de archivo, el enlace para obtenerlo y la última fecha de actualización del archivo. Cada archivo viene en formato CSV, XML y JSON. Como los más actualizados y completos son los CSV, éstos serán los que leeremos.

Ahora lo que hay que comprobar es que la versión del archivo es posterior a la que se encuentra en la base de datos. Algunos de los archivos de los que sacamos los datos vienen con una codificación de caracteres y otros archivos vienen en otra codificación, por lo que hay que comprobar la codificación antes de insertar los datos en la base de datos.

4.4 Problemas encontrados

A medida que se avanzaba con el proyecto se encontraron varios problemas con la fuente de los datos, tanto problemas con el acceso, como problemas con la codificación de los archivos o fallos en los valores de los propios datos como veremos a continuación.

En primer lugar, cuando se empezó a desarrollar el script, el certificado de la página a partir de la cual extraemos los datos no era válido, por lo que se tuvo que definir un contexto que añadiera el certificado de la página web, actualmente el certificado es válido por lo que no hace falta este contexto personalizado, de todas formas se va a mantener por si volviera a ocurrir que tienen un certificado no válido en el futuro.

```
// Para confiar en el certificado de tenerifedata.com
$arrContextOptions= [
    'ssl' => [
        'cafile' => '/home/alberto/tenerifedataTurismoBBDD/cacert.pem',
        'verify peer'=> true,
        'verify peer name'=> true,
    ],
];
```

Figura 11: Creación de un contexto personalizado, al archivo cacert.pem se le incluyó el certificado necesario para que el script pudiera acceder a los datos.

Otro problema encontrado fue que algunos archivos tenían errores en las tildes, pero no parecía ser un fallo de codificación ya que el mismo carácter con tilde lo podía poner de varias formas diferentes, por ejemplo con el nombre Canadá:

1985	4	Canad?1	22	742
1985	4	Canad?10	963	1681
1985	4	Canad?10	4	3
1985	4	Canad?10	24	620
1985	4	Canad?11	0	340
1985	4	Canad?11	0	155
1985	4	Canad?11	0	766
1985	4	Canad?11	0	357
1985	4	Canad?11	3	56
1985	4	Canad?12	20	69
1985	4	Canad?12	0	62
1985	4	Canad?12	6	71
1985	4	Canad?13	9	47

Figura 12: La tilde presentaba problemas en algunos archivos, como por ejemplo con el nombre Canadá.

Para arreglarlo se usaron expresiones regulares con las palabras que daban fallos. Con un array de parejas tenemos por un lado la expresión regular y por otro la cadena de caracteres por la que se sustituirán los nombres de los países con caracteres raros.

```
$arrWrongEncodedWords = array(
    ['/^Espa.*/',"España"],
    ['/^B.*gica/',"Bélgica"],
    ['/^Gran Breta.*/',"Gran Bretaña"],
    ['/^Canad.*/',"Canadá"],
    ['/^Resto de Am.*/',"Resto de América"],
    ['/^Pa.*del Este/',"Países del Este"],
);
```

Figura 13: Expresiones regulares utilizadas para arreglar los fallos de las tildes.

Otro más de los problemas encontrado fue a la hora de leer los archivos de gran tamaño. Al principio se usó una función que leía los archivos con un puntero que apuntaba a la línea actual que se estaba leyendo, pero no llegaba a leer el archivo entero, por esto se cambió la forma de leer los archivos. Almacenando el archivo en su totalidad en una variable en vez de usando un puntero se solucionó el problema, esto implica un mayor gasto de la capacidad de memoria.

Cada línea se pasa como argumento con la función “str_getcsv()” que genera un array con cada valor, teniendo en cuenta que van separados por “,”. Y ahora con expresiones regulares se comprueba si se trata de una cadena de caracteres o de un número. Los números decimales vienen separados por comas (envueltos en comillas dobles para diferenciar la coma decimal de la que separa los valores del archivo CSV), por lo tanto se debe cambiar esa coma decimal por un punto y así poder ingresarse en la base de datos.

También hay archivos en los que cuando se quiere poner un valor nulo, simplemente no se pone nada, pero la base de datos necesita un valor, aunque sea NULL. Se decidió que cuando en uno de los campos del archivo CSV no hubiera nada, se pondría un “0” en la base de datos.

2019	12	65,03	0	40,69
2019	13	59,81	0	37,11
2020	1	61,6	0	38,5
2020	2	71,79	0	44,87
2020	3	31,01	0	19,38
2020	7	51,38		51,38
2020	8	56,04		56,04
2020	9	35,19		35,19
2020	10	39,72		39,72
2020	11	34,61		34,61
2020	12	31,07		31,07
2020	13	46,34	0	35,7
2021	1	23,62		23,62
2021	2	35,54		35,54
2021	3	39,62	0	39,47
2021	4	27,7	0	27,6

Figura 14: Ejemplo de como algunos valores no se daban.

Otro fallo encontrado fue que en uno de los archivos los números tenían puntos de separador de millares.

El último fallo que se encontró fue que un valor en una tabla que daba valores porcentuales salía de rango, por lo que debía ser un error:

59,39	42,7	52,91
62,72	56,64	60,26
77,47	67,37	73,39
73,83	59,34	67,97
76,61	60,86	70,24
78,5	65,35	73,19
76,18	78,07	76,94
76,3	6242	70,79
85,21	71	79,1
89,82	73,79	82,92
82,04	77,48	80,08
71,75	57,66	65,68
72,98	46,39	61,54

Figura 15: Error de un número que debería ser menor de 100.

```
preg_match('/^"?([0-9]+),([0-9]+)"?$/',$data[$c],$re1); // Para cambiar "," decimal por "."
preg_match('/^"?([0-9]+)"?$/',$data[$c],$re2);          // Para números enteros
preg_match('/^(\d{1,3})(\.(\d{3}))+$/',$data[$c],$re3); // Para numeros con "." en el separador
                                                        //de millares
```

Figura 16: Expresiones regulares para arreglar diversos fallos.

Para arreglarlo hubo que especificar que para este archivo (porque no todos dan valores porcentuales), si un valor era mayor que 100 se divide entre 10 hasta que sea menor que 100. Este fallo se encontró cuando ya se había desarrollado la aplicación, ya que cuando consultabas la tabla se veía en el gráfico como un pico fuera de lo normal.

A lo largo del script también se imprime información sobre qué tablas se han actualizado y qué errores se han dado en un log, así tenemos un historial de lo que ocurre cuando se ejecuta.

4.5 Configuración adicional de la base de datos

Para conectarnos a la base de datos desde la aplicación de Android necesitamos un usuario que pueda conectarse desde una IP que no sea localhost. Por lo que creamos un usuario con permisos de SELECT (lectura) que se pueda conectar desde '%', esto indica que puede conectarse desde cualquier IP. También hace falta que abramos el puerto 3306 que es el puerto predeterminado de las bases de datos MySQL.

Con esto ya tenemos la base de datos lista para que nuestra aplicación pueda consultar información en ella.

Capítulo 5 Desarrollo de la aplicación para Android

5.1 Planteamiento de la aplicación

Se pretende que la aplicación muestre los datos de forma comprensible además de información interesante como podría ser la media, mediana, moda y desviación típica de los datos.

El funcionamiento de la aplicación sería el siguiente:

1. Conectarse automáticamente a la base de datos para obtener la lista de tablas.
2. Que el usuario elija la tabla que quiere consultar.
3. El usuario elije que dato quiere observar.
4. Se muestra una gráfica e información interesante.
5. Puede elegir otro dato a observar o volver atrás para seleccionar otra tabla.

5.2 Desarrollo de la aplicación

Las aplicaciones de Android funcionan por clases que se extienden de la clase "Activity". Además se les asigna un "Layout", que es un archivo XML que define la interfaz de usuario. Por otra parte, para almacenar las tablas se cuenta con una clase "Tabla" que contará con el nombre de la tabla, a que conjunto de datos pertenece, los nombres de sus columnas, si la tabla diferencia si los turistas vienen de un país concreto o no y los propios valores de la tabla.

5.2.1 Conexión a la base de datos

Al contrario que con PHP, en el lenguaje Java es necesario descargar una librería para poder conectarte a bases de datos de MySQL. La última versión de la librería oficial no funcionaba por lo que se tuvo que utilizar una más antigua, la librería es mysql-connector-java-3.0.17-ga-bin [8]. Para realizar conexiones en una aplicación para Android, como por ejemplo a una base de datos, se debe crear un "AsyncTask", que se ejecute de forma paralela. Como lo que queremos es que el usuario elija que tabla quiere observar, crearé un botón por cada tabla.

Al crear la clase que se extiende de Activity no sabemos la cantidad de tablas que hay en la base de datos, por lo que hay que esperar a que la tarea paralela, que se conecta a la base de datos, termine. Esto se consigue implementando una función de una interfaz que se pase por parámetro cuando crees la "AsyncTask" dentro de la clase que se extiende de Activity. Y en la función de la interfaz indicamos lo que queremos que se ejecute al final. Por otro lado en la AsyncTask, sobrescribimos el método "onPostExecute()", que como indica su nombre se ejecuta después de terminar la tarea paralela, indicando que se ejecute la función de la interfaz que se le pasó por parámetro en el constructor.

Un ejemplo simplificado sería:

```
public class AuxActivity extends Activity {
    public static ArrayList<String> tablesList;

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_aux);

        AuxAsyncTask auxAsyncTask = new AuxAsyncTask(new AsyncResponse() {
            @Override
            public void processFinish(){
                // Aquí creamos un botón para cada tabla porque esta función la
                // llamaremos cuando termina de ejecutarse auxAsyncTask
            }
        });
        auxAsyncTask.execute();
    }
}

public class AuxAsyncTask extends AsyncTask<Void, Void, String>{
    public AsyncResponse delegate = null;

    public AuxAsyncTask(AsyncResponse asyncResponse){
        this.delegate = asyncResponse;
    }

    protected String doInBackground(Void… params){
        // Conectarse a la base de datos y pasar los nombres de las tablas a la Activity.
        // …
        // Gracias a ésto pasamos los nombres de las tablas a AuxActivity
        AuxActivity.tablesList.add(nombreDeLaTabla);

        return “Complete”;
    }

    protected void onPostExecute(String result){
        if(result.equals(“Complete”){
            delegate.processFinish();
            // Gracias a esto se añaden los botones solo después de terminar
            // la consulta a la base de datos.
        }
    }
}

public interface AsyncResponse {
    void processFinish();
}
```

Figura 17: Código para ejemplificar AsyncTasks

5.2.2 Elección de la tabla a consultar

Para que el usuario elija la tabla que desea consultar vamos a crear un botón por cada tabla. Tenemos un ArrayList con los objetos que representan las tablas, por lo que lo recorremos y para cada una creamos un botón en el Layout (archivo XML) de la Activity, configurando para cada uno un "OnClickListener" que nos lleve a la siguiente activity y transmitiendo que tabla se ha elegido.

A continuación un ejemplo simplificado de como se crea cada botón, este bloque de código se tiene que repetir por cada botón que se quiera crear:

```
// Para crear un botón, creamos un objeto Button de la siguiente forma
Button btn = new Button(this);

// Después si queremos configurar el OnClickListener haríamos lo siguiente
btn.setOnClickListener(new View.OnClickListener {
    @Override
    public void onClick(View v){
        //...
    }
});

// Ahora solo queda situarlo dentro del Layout, para ello necesitamos
// un ConstraintSet, unos guidelines que definimos en el Layout para
// que no se quede el botón pegado a los bordes de la pantalla y
// además necesitaremos una lista a la que ir añadiendo los
// botones (contactButtonList).
ConstraintSet set = new ConstraintSet();
// Ajusta el ancho
set.constrainDefaultWidth(btn.getId(),
                          ConstraintSet.MATCH_CONSTRAINT_SPREAD);
// Ajusta la distancia a la parte izquierda de la pantalla
set.connect(btn.getId(), ConstraintSet.LEFT, R.id.guideline19,
            ConstraintSet.RIGHT, 8);
// Ajusta la distancia a la parte derecha de la pantalla
set.connect(btn.getId(), ConstraintSet.RIGHT, R.id.guideline20,
            ConstraintSet.RIGHT, 8);
// Si es el primer botón se ajusta a la parte de arriba de la
// pantalla, si no lo es, se ajusta al último botón que ha sido
// ajustado
if (contactButtonList.isEmpty()) {
    set.connect(btn.getId(), ConstraintSet.TOP, R.id.guideline18,
                ConstraintSet.BOTTOM, 8);
} else {
    set.connect(btn.getId(), ConstraintSet.TOP, contactButtonList.get(
        contactButtonList.size() - 1).getId(), ConstraintSet.BOTTOM);
}

contactButtonList.add(btn);
```

Figura 18: Ejemplo de como crear un Button y agregarlo al Layout.

Como hay muchas tablas, vamos a crear una "Activity" intermedia que pregunte al usuario el conjunto de datos que quiere consultar, y luego qué tabla de ese conjunto de datos. Para esto agregué en la base de datos a que conjunto pertenecía cada tabla y luego en la aplicación se recorre la lista de tablas y se crea un botón por cada conjunto que no se haya encontrado en otra tabla anterior.

Figura 19: Captura sobre la selección de conjunto de datos

Figura 20: Captura sobre la elección de tablas

Cuando el usuario elige un conjunto de datos, se lanza la siguiente clase Activity pasando como argumento el conjunto elegido y la lista de tablas, recorremos la lista de objetos tipo Tabla para crear un botón para cada tabla que pertenece al conjunto de datos escogido. En esta nueva clase Activity crearemos los botones de igual manera que en la clase Activity anterior.

5.2.3 Elección del dato a mostrar.

Una vez el usuario ha escogido una tabla, se pasa a la última Activity, en ella tendrá que decidir qué dato mostrar y para ello tenemos dos desplegables, uno para el dato en concreto y otro para la nación de los turistas a los que se tiene en cuenta, no todas las tablas diferencian por nacionalidad a los turistas. También habrá una barra para escoger el rango de los años a mostrar. Cuando ya se haya hecho una selección hay que darle al botón “Añadir”, este botón se crea directamente en el Layout ya que siempre será un solo botón, previamente no sabíamos la cantidad exacta de botones que necesitábamos y por eso se creaban dentro del código Java. Lo único que hay que hacer en el código Java con el botón es configurar OnClickListener para definir que acción se ejecuta cuando es presionado.

5.2.4 Mostrar los datos

Una vez se presiona el botón “Añadir”, se muestran los datos en una gráfica. Para mostrar la gráfica se ha utilizado la librería MPAndroidChart [9]. Lo que queremos mostrar es una gráfica lineal, para ello en el Layout añadimos un LineChart. Luego en el código Java, a este LineChart le añadimos un ArrayList de IlineDataSet. Esto es porque cada IlineDataSet representa una conjunto de puntos en el gráfico ya que la librería permite mostrar varias a la vez y vamos a aprovecharlo. Por eso con el botón previamente mencionado, en el que pone “Añadir”, podremos mostrar varios datos a la vez para compararlos visualmente. Para diferenciar entre ellos existe una leyenda debajo del gráfico que dice de que color es cada dato mostrado. Por último, también se agregó la opción de pulsar sobre un punto del gráfico para que aparezca un recuadro que muestre los valores “x” e “y”.

A parte del gráfico también queremos mostrar infomración adicional como la media, desviación típica, mediana y moda. Para ello se creó una clase a llamada SimpleMath que tiene las funciones necesarias para calcular estos datos, simplemente recibe un array con los datos deseados y la función devuelve el resultado correspondiente. Se pueden mostrar datos a la vez en el gráfico, pero para esta información adicional solo mostraremos la del último dato consultado, ya que no tenemos suficiente espacio en la pantalla.

En las próximas páginas mostramos varios ejemplos:

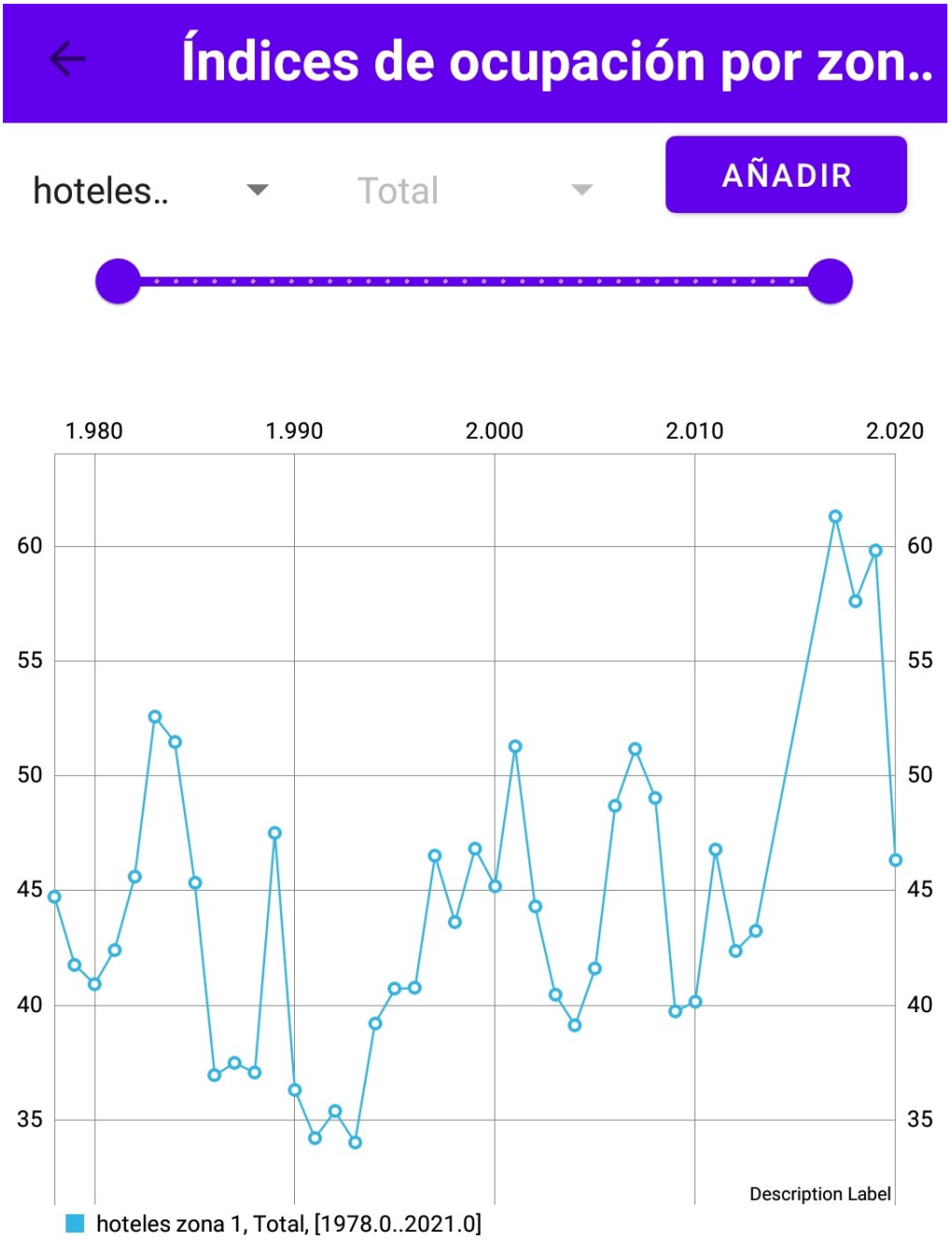

Figura 21: Captura de un gráfico lineal. Ejemplo de una sola consulta.

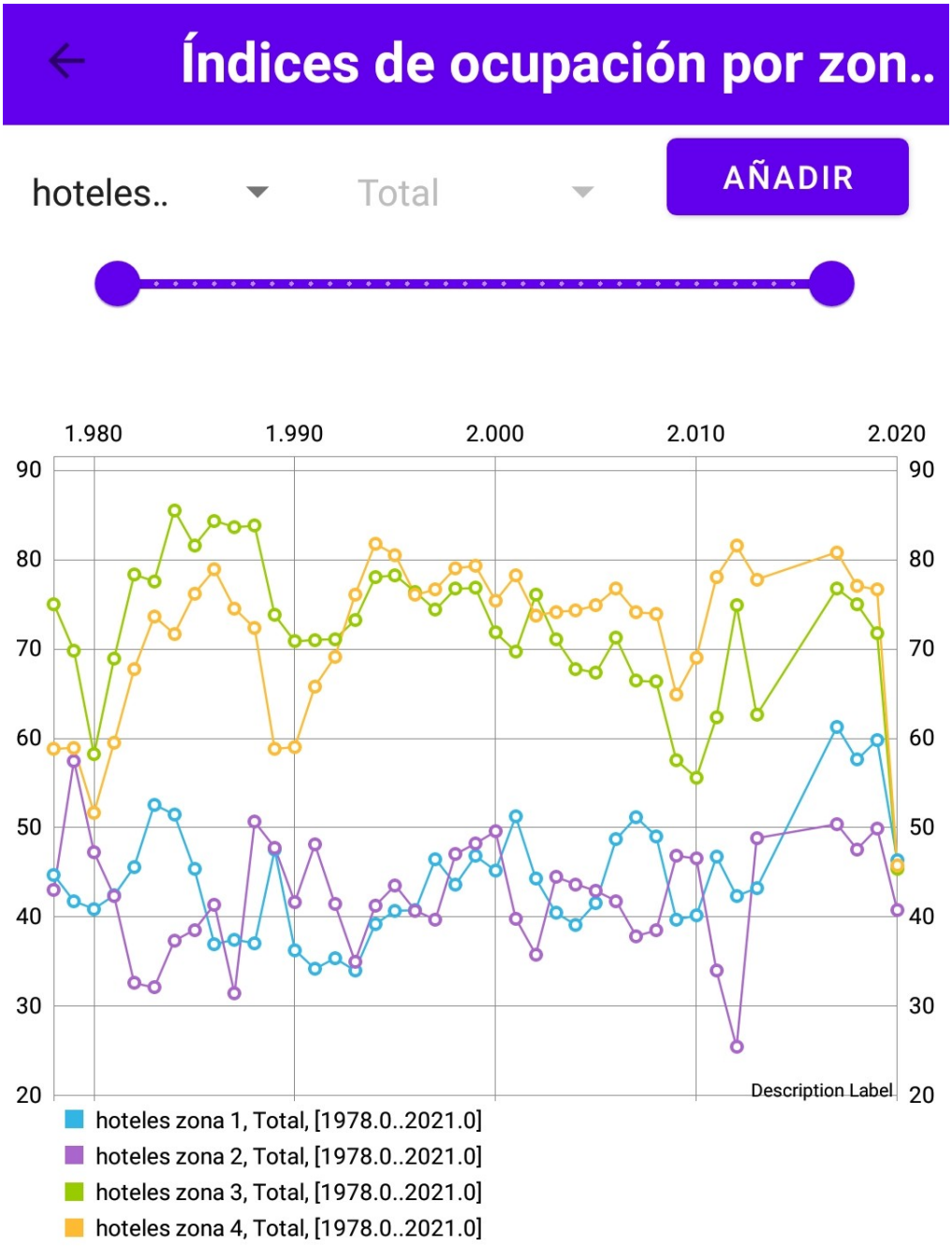

Figura 22: Captura de un gráfico lineal. Ejemplo con varias consultas juntas.

Media	Moda
7,920	Tiende a distribución uniforme discreta
Desviación típica	Mediana
0,227	7,900

Figura 23: Captura de un gráfico lineal. Ejemplo con recuadro que describe un punto en el gráfico.

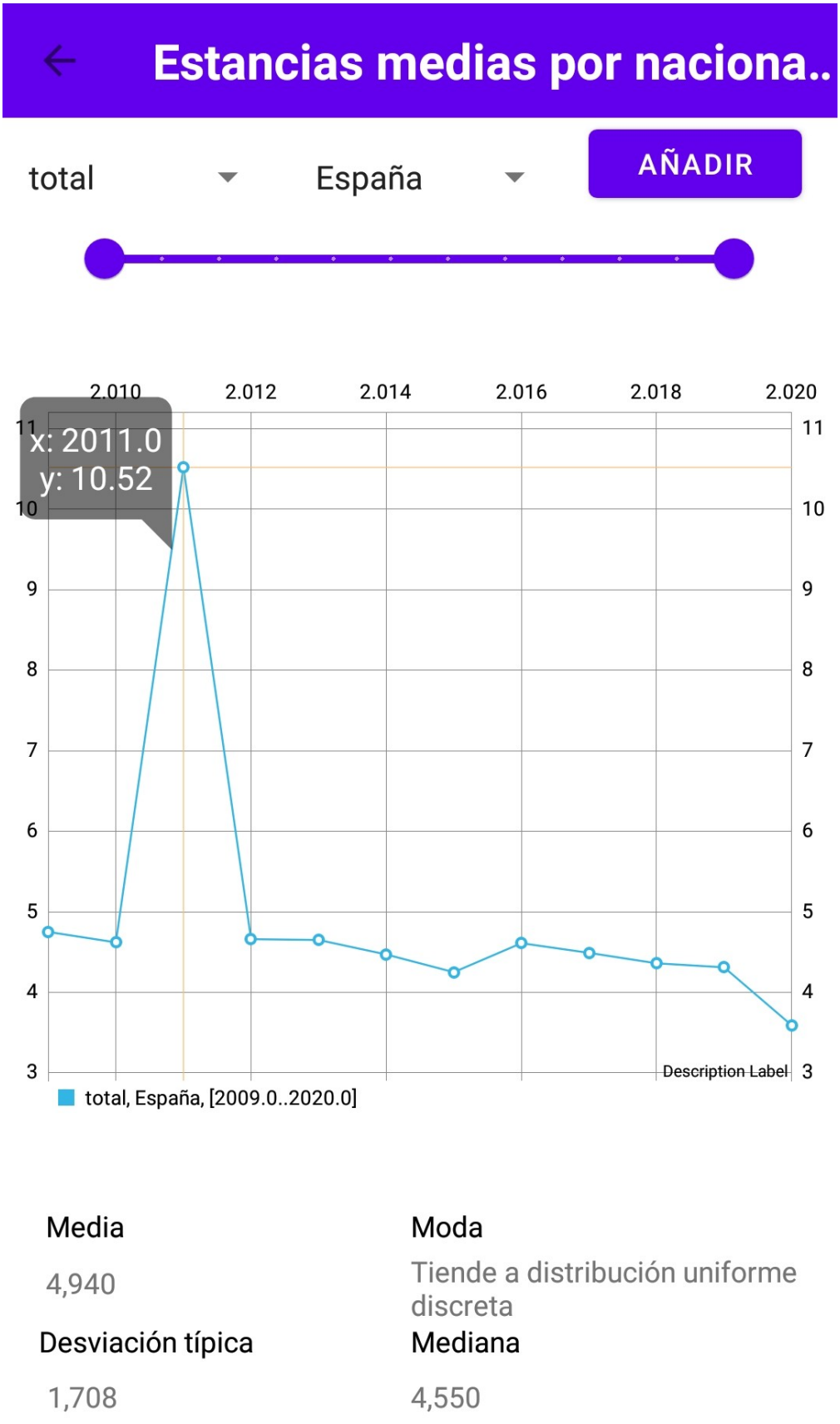

Figura 24: Captura de un gráfico lineal. Ejemplo en el que el recuadro se saldría del gráfico.

5.2.5 Añadidos visuales

Para que la aplicación tuviera una mejor apariencia se creó una pantalla inicial con el nombre de la aplicación y el logo de la página de la que se extraen los datos, esta pantalla se muestra durante 2 segundos.

El color azul seleccionado para la pantalla inicial y el logo es el mismo azul que se utiliza en la página web de la que se extraen los datos, dentro de la aplicación se utiliza un violeta que combina con el azul seleccionado.

Otra mejora visual, como se puede ver en ejemplos anteriores, es que se incluyeron las barras de la parte superior de la pantalla (llamadas Toolbar), están formadas por el título de la pantalla actual y una flecha que te devuelve a la pantalla anterior.

Pantalla inicial de la aplicación:

Figura 25: Captura del inicio de la aplicación.

Para finalizar también diseñé el logo de la aplicación:

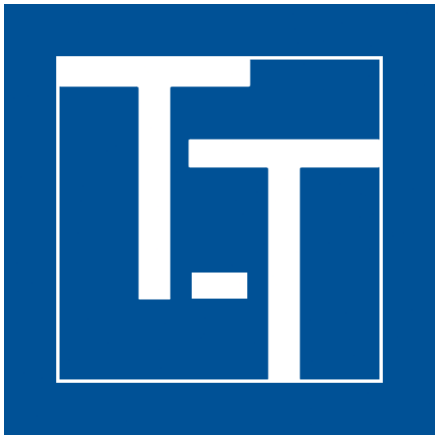

Figura 26: Logo de la aplicación.

5.3 Esquema final

Esquema simplificado del funcionamiento del proyecto. Primero el script PHP comprueba que existan nuevos datos en la página web de origen, si existen nuevos datos actualiza las tablas necesarias. Durante este proceso también se arreglan todos los fallos presentes en los datos originales. Por otro lado, los dispositivos Android que tengan la aplicación instalada, podrán usarla para consultar la base de datos a través de la interfaz gráfica de usuario (GUI) con la que cuenta.

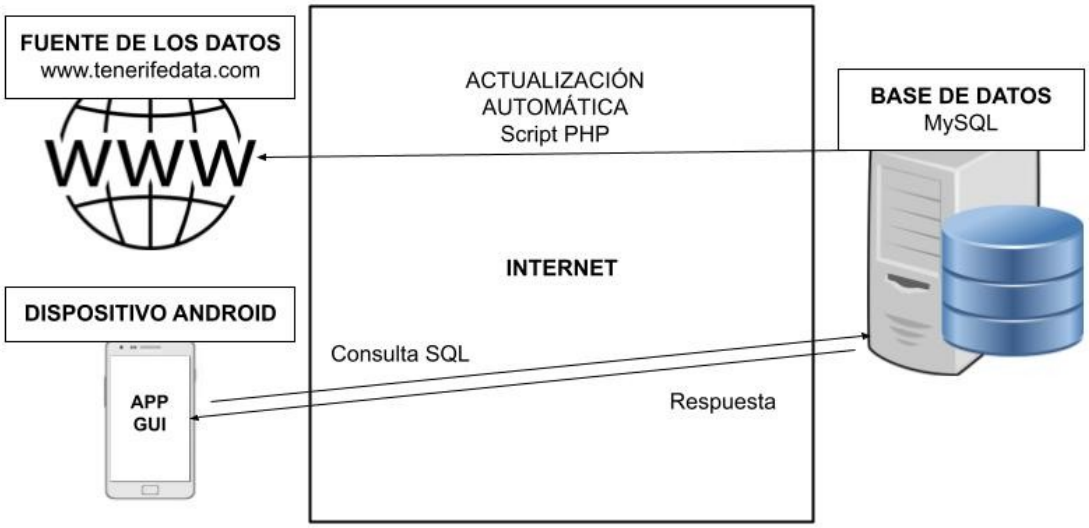

Figura 27: Esquema simplificado del funcionamiento del proyecto.

Capítulo 6 Conclusiones y líneas futuras

6.1 Conclusiones

Partiendo de los datos de la página web TenerifeData, hemos obtenidos los datos, los hemos corregido y trasladado a una base de datos personal y diseñamos una aplicación para Android. Es una aplicación funcional, la cual ha supuesto el reto de ser la primera aplicación para Android que desarrollabamos, ha resultado desafiante e interesante.

Todo eso se traduce en que se ha trabajado con lenguaje SQL para la base de datos, lenguaje PHP para un script que actualice la base de datos, lenguaje Java para el desarrollo del código de la aplicación, archivos XML que definían las interfaz de usuario de la aplicación y por último se ha realizado el diseño gráfico con la herramienta Gimp.

Esto engloba bastantes facetas que se nos han inculcado a lo largo de la carrera, tanto las asignaturas de desarrollo y análisis de código, como el desarrollo y diseño de bases de datos, el desarrollo de lenguaje de marca y diseño gráfico.

Para desarrollar este Trabajo de Fin de Grado también me ayudó mucho mi estancia como estudiante de prácticas externas en Metrotenerife S.A., ya que allí desarrollé una aplicación web con el lenguaje Java gracias al framework Spring MVC. La aplicación web se conectaba a una base de datos de la empresa y se podían realizar consultas y modificaciones a la base de datos. Por lo que las diferencias son que usé lenguaje de marca XML en lugar de HTML, y en lugar de aprender a desarrollar para Android, aprendí a desarrollar una aplicación con Spring MVC.

6.2 Líneas futuras

De cara al futuro de la aplicación, el mayor inconveniente que presenta es el siguiente: la base de datos con la que se trabaja se encuentra alojada en un ordenador personal y no es aconsejable hacer que sea accesible desde cualquier parte del mundo para consultar la base de datos. Se debería crear un servidor dedicado para la base de datos, por ejemplo en la nube, que además elimina también el problema de tener que dejar encendido dicho ordenador personal, a parte del mayor problema que implica exponerlo a que sea accesible a cualquiera. Para esto se podría utilizar el servicio que ofrece DigitalOcean[10] llamado Managed Databases, o el servicio de Google llamado Cloud SQL[11], te permiten tener una base de datos MySQL accesible a través de internet.

Un cambio muy notable sería el de mejorar la estética de la aplicación, por ejemplo cuando se va a elegir qué tabla quieres consultar, se utiliza una lista de botones con los nombres de las tablas, cumple la función pero se podría mejorar. También la pantalla en la que se muestra la gráfica y los datos de interés está bastante saturada de elementos.

Otras posibles mejoras serían añadir más funcionalidad a la aplicación, como añadir descripciones más detalladas de los conjuntos de datos que se están accediendo, aunque el propio nombre de cada uno ya es descriptivo, pero en menor medida. O por ejemplo, una mayor funcionalidad sería expandir la aplicación para que muestre los datos de toda la página web TenerifeData, de la que extraemos los datos, ya que tiene datos sobre demografía, movilidad, mercado laboral, comercio o bienestar social entre otros.

Capítulo 7 Summary and Conclusions

7.1 Conclusions

Starting off with data from the web page TenerifeData, we obtained the data, corrected it and transferred it to our own database and then we designed an application for Android. It's a functional application, which has supposed the challenge of being the first Android application that we have developed and has been really interesting.

All this is also translated as a project which has had SQL language to develop the database, a PHP script that updates said database, Java programming language for the Android application code, XML files that define the user interface and last, the design of the application's logo and main screen with the tool Gimp.

This encompasses many facets that have been inculcated throughout our career, we have development and analysis of code, development and design of databases, development of markup language and also graphic design.

To develop this project, my time at Metrotenerife S.A. (where I was doing my mandatory external practice sessions) really helped me. I had to develop a web application that made queries to the company’s database through formularies that the employees had to fill. So there I had to learn how to develop a web application thanks to the Spring MVC framework, but now I developed an application for Android, and also I wrote in HTML for the web application but now I was working with XML for the application user interface.

7.2 Future work

In terms of the future of the application, the major inconvenience is the following: the database that we work with is allocated in a personal computer and isn't accessible from outside our private network, nor to read and even less to change the data because it isn't advised to make the necessary changes in a personal computer. To change this we would have to create a dedicated server for the database, in the cloud for example, that also solves the problem of having to leave your personal computer always on. Some examples that can solve this problem are DigitalOcean's [10] service called Managed Databases, or Google's [11] service called Cloud SQL, they allow you to have a database accessible from the internet.

Another notable change would be to improve the application's aesthetic, for example, when the user chooses which data he wants to consult, the application shows a list of buttons with table names on them, it works but it can be improved upon. Also the screen that shows the line chart and the interesting information is really saturated with elements.

Lastly, other possible upgrades are adding more functionality to the application, like adding descriptions of every table that the user can consult, the name of the table isn't always descriptive enough. Or for example, another improvement would be to expand the application to all the data that is available at TenerifeData, the page has information about demographics, mobility, labour market, trade or social welfare among others.

Capítulo 8 Presupuesto

8.1 Desarrollo del software

Se tiene en cuenta que la planificación tiene un valor de 30€/h y el desarrollo un valor de 23€/h.

Tipos	**Planificación(h)**	**Desarrollo(h)**	**Total(h)**	**Coste(€)**
Base de datos	20	30	50	1290
Aplicación	20	40	60	1520
Diseño gráfico	5	5	10	265
Total	45	75	120	3075

Tabla 1: Coste de desarrollo del software del proyecto.

8.2 Recursos para el desarrollo

Se ha optado por la utilización de software gratuito como MySQL, AndroidStudio, Gimp o LibreOffice ya que son herramientas de buena calidad por lo que no creemos necesario la utilización de herramientas de pago como podría ser Adobe Photoshop. En el caso de que se quiera utilizar un servidor en la nube para alojar la base de datos, no se ha encontrado ninguna opción gratuita, la más factible que se ha encontrado es DigitalOcean Managed Databases [10].

Tipos	**Coste(€/mes)**
Android Studio	0
Gimp	0
MySQL Server	0
DigitalOcean Managed Databases	15

Tabla 2: Recursos que se necesitarán para el desarrollo del proyecto.

8.3 Gastos totales

Tipos	Coste plano(€)	Coste Mensual(€)
Desarrollo del Software	3075	
Recursos para el desarrollo		15

Tabla 3: Resumen de gastos total

Bibliografía

[1] TenerifeData,
https://www.tenerifedata.com

[2] Díaz, A. (2021, 3 de marzo). *El turismo en Canarias.* Statista.,
https://es.statista.com/temas/4115/el-turismo-en-canarias/

[3] *What is Ubuntu? - Definition from WhatIs.com*. (2009, 15 de junio). SearchDataCenter.
https://searchdatacenter.techtarget.com/definition/Ubuntu

[4] Gustavo. (s. f.). *¿Qué Es MySQL? Explicación Detallada Para Principiantes*. Tutoriales Hostinger.
https://www.hostinger.es/tutoriales/que-es-mysql

[5] Shafirov, M. (2017, 17 de mayo). *Kotlin on Android. Now official.* JetBrains Blog.,
https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/

[6] Miller, J. (s. f.). Which Version Control Should Developers Use? | BairesDev. BairesDev.
https://www.bairesdev.com/blog/which-version-control-system-developers-use/

[7] GIMP - About GIMP. (s. f.). GIMP - GNU Image Manipulation Program.
https://www.gimp.org/about/introduction.html

[8] Borau, C. (2015, 23 de noviembre). Android JDBC mysql java connector. Stack Overflow.,
https://stackoverflow.com/a/33874206

[9] Jay, P. (2019, 20 de marzo). *GitHub MPAndroidChart: A powerful Android chart view / graph view library.* GitHub.,
https://github.com/PhilJay/MPAndroidChart

[10] DigitalOceans Managed Databases,
https://www.digitalocean.com/products/managed-databases/

[11] Google's Cloud SQL,
https://cloud.google.com/sql