



Trabajo de Fin de Grado

Implementación de Apps en Android

Android Apps Implementation

Christian Ramos Barrera

La Laguna, 30 de junio de 2021

D. **Alejandro Pérez Nava**, con N.I.F. 43821179S profesor asociado de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

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

Que la presente memoria titulada:

"Implementación de Apps en Android"

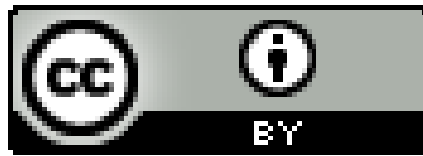
ha sido realizada bajo su dirección por D. **Christian Ramos Barrera**, con N.I.F. 79089748Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 30 de junio de 2021

Agradecimientos

Quiero agradecer a mi familia, que me ha dedicado todo su tiempo, todo su esfuerzo y todos sus recursos con tal de educarme y formarme lo mejor posible para afrontar la vida. También a todos los profesores que he tenido, tanto en la universidad como fuera de ella, por haber fomentado, aunque algunos más que otros, el desarrollo de mi curiosidad y conocimiento. También quiero agradecer a aquellos que me han formado como profesional y como persona, realizando un trabajo que nunca se podrá valorar lo suficiente.

Licencia



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

Resumen

El objetivo de este proyecto ha sido desarrollar una aplicación para Android, la cual, consiste en un buscador de eventos culturales en las Islas Canarias. Los datos a utilizar se encuentran publicados en una página web(1) y fueron subidos por la Consejería de Educación, Universidades, Cultura y Deportes del Gobierno de Canarias.

Para el correcto funcionamiento de la aplicación se han implementado los siguientes componentes:

- *Base de Datos - MySQL*
- *Actualización automática de la Base de Datos - Python*
- *Conexión y WebServices - XAMPP(PHP)*
- *Aplicación Cliente - Android Studio(Java)*

Además de la búsqueda se ha implementado la opción de eventos mas visitados. Mostrando en la pantalla principal los 6 eventos más consultados por los usuarios.

Palabras clave: Android, eventos culturales, MySQL, WebServices, PHP, Java

Abstract

The objective of this project has been to develop an application for Android, which consists of a search engine for cultural events in the Canary Islands. The data to be used are published on a web page(1) and were uploaded by the Ministry of Education, Universities, Culture and Sports of the Government of the Canary Islands.

For the correct operation of the application, the following components have been implemented:

- *Database - MySQL*
- *Automatic Database Update - Python*
- *Connection and WebServices - XAMPP(PHP)*
- *Client Application - Android Studio(Java)*

In addition to the search, the option of most visited events has been implemented. Showing on the main screen the 6 events most consulted by users.

Keywords: Android, cultural events, MySQL, WebServices, PHP, Java

Índice general

1. Introducción	1
1.1. Problema a resolver	1
1.2. Antecedentes	1
1.2.1. Tomaticket(2)	2
1.2.2. Buscametas(4)	2
1.2.3. Google	2
1.3. Tendencia del mercado	2
1.4. Fases de desarrollo	2
1.5. Estructura del documento	3
2. Tecnologías	4
2.1. MySQL	4
2.2. Sublime Text	5
2.3. XAMPP	5
2.4. Android Studio	5
2.5. Github	6
2.6. Latex	6
3. Desarrollo	8
3.1. BackEnd	8
3.1.1. Análisis de los datos a tratar	8
3.1.2. Base de Datos - MySQL	9
3.1.3. Actualización automática	10
3.1.4. XAMPP	10
3.1.5. WebServices	11
3.2. FrontEnd	16
3.2.1. Página principal	16
3.2.2. Buscador	17
3.2.3. Lista de eventos	18
3.2.4. Detalles del evento	19
3.2.5. Esquema final	20
4. Despliegue	21
4.1. Modificaciones necesarias	21
5. Conclusiones y líneas futuras	22
5.1. Conclusiones	22
5.2. Líneas Futuras	22

6. Summary and Conclusions	23
6.1. Conclusions	23
6.2. Summary	23
7. Presupuesto	24
7.1. Desarrollo Software	24
7.2. Infraestructura Hardware	24
7.3. Total	25

Índice de figuras

3.1. Base de Datos Web	8
3.2. Líneas de ejemplo	9
3.3. Base de Datos final	9
3.4. Código para insertar datos	10
3.5. Código para actualización automática	10
3.6. XAMPP panel de control	11
3.7. XAMPP panel de control Web	11
3.8. conexion.php	12
3.9. buscar_evento.php - Cabecera	13
3.10 buscar_evento.php - Parsing	13
3.11 buscar_evento.php - Final	14
3.12 Ejemplo de Salida - Web	14
3.13 top_events.php	15
3.14 event_click.php	15
3.15 Página Principal	16
3.16 Buscador	17
3.17 Lista de eventos	18
3.18 Detalles del evento	19
3.19 Esquema de los componentes	20

Índice de tablas

7.1. Resumen de Gastos - Desarrollo	24
7.2. Resumen de Gastos - Hosting	25
7.3. Resumen de Gastos - Total	25

Capítulo 1

Introducción

1.1. Problema a resolver

En Canarias se organizan un gran número de eventos prácticamente a diario, pese a ello, muchas veces optamos por realizar las mismas actividades con las que ya estamos familiarizados. Esta situación suele darse muchas veces no por falta de ganas de realizar otras actividades, si no, por el hecho de no conocer todos los eventos organizados incluso si estos se desarrollan en las cercanías o son de modalidad online.

Para solucionar este problema se plantea la creación de una aplicación móvil que englobe todos estos eventos y el usuario pueda filtrarlos por medio de un buscador.

1.2. Antecedentes

La consulta de datos siempre ha sido un problema muy analizado y trabajado a lo largo de la historia. Por lo general, consiste en un usuario quiere extraer un registro o dato según unos requisitos o necesidades, para en base a ello, tomar decisiones u obtener información de esos datos. Esto abarca desde, buscar un libro en una biblioteca basándose en un género en concreto a consultar en una web los pisos en alquiler cerca de tu zona.

En la actualidad se intenta hacer estas consultas lo más intuitivas posible, por medio de interfaces que luego son traducidas a los comandos equivalentes, logrando así no necesitar de un experto que conozca los parámetros o gramática exacta para acceder a estos datos. Pudiendo ser los propios usuarios quienes obtengan la información que necesitan logrando acelerar el proceso.

A día de hoy no existe una aplicación móvil que cubra la necesidad planteada en concreto. Pero existen múltiples páginas web que recogen eventos a nivel nacional, estas además, son capaces de hacer reservas gracias al pago online.

1.2.1. Tomaticket(2)

Esta página maneja una gran variedad de tipos de eventos, desde fiestas y cine hasta cursos y deportes, incluso participan con ONG's. Trabaja a nivel nacional aunque en canarias tiene mucha presencia en festivales importantes. Además ofrece la posibilidad de crear tu propio evento para captar participantes a través de su página.

1.2.2. Buscametas(4)

Página destinada a los eventos deportivos a nivel nacional, capaz de filtrar por tipos de deportes como running, trail, ciclismo, etc. También cuenta con una sección de noticias deportivas redactadas por la propia página, con análisis de la participación y desempeño en los eventos.

1.2.3. Google

Google ha desarrollado una función para búsqueda de eventos muy completa, ya que, es capaz de mostrarte las distintas páginas donde acceder a las entradas del mismo, guardar el evento e incluso enlazarlo con el google calendar. Con simplemente buscar en google.com "eventos canarias", puedes desplegar esta funcionalidad.

1.3. Tendencia del mercado

A día de hoy este tipo de aplicaciones suelen ser desplegadas en una página web, por su fácil montaje y para facilitar que cualquier persona con acceso a internet pueda disfrutar de su servicio. La aplicaciones desarrolladas para dispositivos móviles suelen requerir de un seguimiento más estricto del usuario y contar con un sistema de logros que el usuario puede consultar rápidamente o incluso offline.

También existen muchas aplicaciones que no son desarrolladas específicamente para el dispositivo móvil. Primero son desarrolladas para una página web y después se encapsulan en una APK para facilitar al usuario el rápido acceso.

1.4. Fases de desarrollo

Para el desarrollo de este trabajo se ha seguido el siguiente flujo de tareas:

- Crear Base de Datos en MySQL
 - Documentación y primeros pasos
 - Creación de la tabla
 - Inserción de datos
 - Creación del script para actualización automática

- Crear WebServices para poder acceder a los datos con XAMPP
 - Documentación y configuración
 - Creación de scripts para los WebServices
- Implementar Aplicación en Android Studio
 - Creación de mockups
 - Implementar formularios
 - Implementar la conexión con los WebServices
 - Muestreo de Datos
- Redacción de Memoria (Tarea en paralelo a las anteriores)
 - Retoques y Correcciones

1.5. Estructura del documento

- **Capítulo 2** - Tecnologías

Explicación detallada de todas las tecnologías utilizadas a lo largo del desarrollo del proyecto.
- **Capítulo 3** - Desarrollo

Todo el desarrollo del proyecto explicado en orden cronológico y separado en dos grandes campos, Backend (Servidor) y Frontend (Aplicación Android).
- **Capítulo 4** - Despliegue

Cambios requeridos en caso de querer desplegar la aplicación.
- **Capítulo 5** - Conclusiones y líneas futuras

Resultados obtenidos tanto del proyecto como personales y algunos problemas encontrados. Por último una serie de mejoras y cambios que se pueden realizar a la aplicación.
- **Capítulo 6** - Summary and Conclusions

Igual que el capítulo anterior pero redactado en inglés.
- **Capítulo 7** - Presupuesto

Todos los costes del proyecto desglosados y separados entre costes planos y mensuales. Contiene los gastos de desarrollo y hardware.

Capítulo 2

Tecnologías

2.1. MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

Está desarrollado en su mayor parte en **ANSI C** y **C++**. Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo **LAMP** y **WAMP**.

LAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- **Linux**, el sistema operativo.
- Apache, el servidor web.
- MySQL/MariaDB, el gestor de bases de datos.
- PHP, el lenguaje de programación.

WAMP es el acrónimo usado para describir un sistema de infraestructura de Internet que usa las siguientes herramientas:

- **Windows**, el sistema operativo.
- Apache, el servidor web.
- MySQL/MariaDB, el gestor de bases de datos.
- PHP, el lenguaje de programación.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

2.2. Sublime Text

Sublime Text es un editor de texto y editor de código fuente. Está escrito en **C++** y **Python**. Desarrollado originalmente como una extensión de **Vim**, con el tiempo fue creando una identidad propia. Aún conserva un modo de edición tipo vi llamado Vintage mode.

Algunas de sus características más importantes son:

- **Minimapa:** consiste en una previsualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.
- **Soporte nativo para infinidad de lenguajes:** Soporta de forma nativa 43 lenguajes de programación y texto plano.
- **Syntax Highlight configurable:** El remarcado de sintaxis es completamente configurable a través de archivos de configuración del usuario.
- **Configuración total de Keybindings:** Todas las teclas pueden ser sobrescritas a nuestro gusto.

2.3. XAMPP

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. A partir de la versión, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

2.4. Android Studio

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas GNU/Linux, macOS, Microsoft Windows y Google Chrome OS. Ha sido diseñado específicamente para el desarrollo de Android.

Desde el 7 de mayo de 2019, Kotlin es el lenguaje preferido de Google para el desarrollo de aplicaciones de Android. Aun así, Android Studio admite otros lenguajes de programación, como Java y C ++.

Requisitos para la version 3.x:

OS version	Window 10/8/7 (32 o 64 bit)
	Mac OS X 10.10 (Yosemite) o superior, hasta 10.13 (macOS High Sierra)
	Linux GNOME o KDE desktop
RAM	4 GB RAM mínimo, 8 GB RAM recomendado más 1GB adicional para el emulador de Android
Espacio de almacenamiento	2 GB para Android Studio, 4 GB recomendados (500 MB para IDE y al menos 1.5 GB para Android SDK, imágenes de sistema de emulador y cachés)
Java version	Java Development Kit (JDK) 8
Resolución de pantalla	1280x800 mínimo, 1440x900 recomendado

2.5. Github

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en **Ruby on Rails**. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC.

Sus características son:

- Wiki para cada proyecto.
- Página web para cada proyecto.
- Gráfico para ver cómo los desarrolladores trabajan en sus repositorios y bifurcaciones del proyecto.
- Funcionalidades como si se tratase de una red social, por ejemplo, seguidores.
- Herramienta para trabajo colaborativo entre programadores.
- Gestor de proyectos de estilo Kanban.
- Herramientas de CI Codespaces un IDE en la nube para los repositorios.

2.6. Latex

LaTeX es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.

LaTeX está formado por un gran conjunto de macros de **TeX**, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, **TeX**, creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados en **LaTeX**, se considera adecuada a las necesidades de una editorial científica de primera línea, muchas de las cuales ya lo emplean. **LaTeX** es software libre bajo licencia LPPL.

Una de las ventajas de **LaTeX** es que la salida que ofrece es siempre la misma, con independencia del dispositivo (impresora, pantalla, etc.) o el sistema operativo (MS

Windows, MacOS, Unix, distribuciones GNU/Linux, etc.) y puede ser exportado a partir de una misma fuente a numerosos formatos tales como Postscript, PDF, SGML, HTML, RTF, etc. Existen distribuciones e IDEs de LaTeX para todos los sistemas operativos más extendidos, que incluyen todo lo necesario para trabajar. Hay, por ejemplo, programas para Windows como TeXnicCenter, para GNU/Linux como Kile, o para MacOS como TeXShop, todos liberados bajo la Licencia GPL. Existen además los editores multiplataformas (para MacOS, Windows y Unix) Texmaker y TeXworks, que también son liberados bajo licencia GPL.

Capítulo 3

Desarrollo

3.1. BackEnd

Lo primero a desarrollar ha sido el Backend (Base de Datos, webservice, actualizador), ya que, a la hora de comprobar el correcto funcionamiento de los webservices basta con acceder a la ruta donde está nuestro servicio escuchando.

3.1.1. Análisis de los datos a tratar

En primer lugar analicé los tipos de datos en la página(1) a usar:

Columna	Tipo	Etiqueta	Descripción
FECHA_INICIO	timestamp		
FECHA_FIN	timestamp		
DURACION_DIAS	numeric		
TIPO_EVENTO	text		
ISLA	text		
MUNICIPIO	text		
DENOMINACION_ESPACIO	text		
LUGAR	text		
TITULO	text		
DESCRIPCION	text		
HORA	text		
MINUTO	text		
MAS_INFO	text		
URL_IMAGEN	text		

Figura 3.1: Base de Datos Web

Como se puede comprobar, los campos son prácticamente todos de tipo texto, por lo tanto habrá que usar filtros específicos para campos de este tipo.

id	FECHA...	FECHA_...	DURACI...	TIPO_E...	ISLA	MUNICI...	DENOMI...	LUGAR	TITULO	DESCRI...	HORA	MINUTO	MAS_IN...	URL_IM..
1	2021-05-...	2021-05-...	0	Música	Gran Ca...	Las Pal...	Auditorio...		Revolver...	Conciert...	20	30	https://bi...	http://w ^
2	2021-01-...	2021-05-...	118	Exposici...	La Palma	Breña Baja	---	Ayuntam...	Exposici...	Organiz...	--	--	https://bi...	http://w
3	2021-02-...	2021-12-...	315	Literatura	Gran Ca...	Moya	Casa Mu...		Natalia ...	Exposici...	--	--	http://ww...	http://w
4	2021-02-...	2021-05-...	79	Exposici...	Tenerife	Santa Cr...	TEA Ten...		'Escalas ...	Muestra ...	--	--	https://bi...	http://w

Figura 3.2: Lineas de ejemplo

Además se presenta un problema, cuando un campo no tiene valor no se inserta como vacío, se insertan unos guiones '—' para indicarlo. Cosa a tener en cuenta a la hora de realizar las búsquedas, al igual que el uso de tildes y 'ñ'.

3.1.2. Base de Datos - MySQL

Una vez conocidos los datos a manejar paso a la creación de la base de datos en MySQL(5). Es una base de datos de código abierto muy popular, de fácil despliegue y con una amplia variedad de tutoriales disponibles. A parte de estas características, otro motivo para haber elegido MySQL como Base de Datos es la fácil conexión con XAMPP aplicación la cual se entrará en detalles más adelante.

He decidido usar la columna 'MAS_INFO' como clave primaria para la base de datos, ya que, consiste en una url única para cada evento con información adicional.

Por lo tanto despues de ejecutar los comandos correspondientes de CREATE DATABASE <name>y CREATE TABLE Eventos ([<Column>,...]), la estructura de la base de datos es la siguiente:


#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 FECHA_INICIO	date			Sí	NULL
<input type="checkbox"/>	2 FECHA_FIN	date			Sí	NULL
<input type="checkbox"/>	3 DURACION_DIAS	int		UNSIGNED	Sí	NULL
<input type="checkbox"/>	4 TIPO_EVENTO	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	5 ISLA	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	6 MUNICIPIO	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	7 DENOMINACION_ESPACIO	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	8 LUGAR	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	9 TITULO	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	10 DESCRIPCION	varchar(200)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	11 HORA	varchar(10)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	12 MINUTO	varchar(10)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	13 MAS_INFO 	varchar(150)	utf8mb4_spanish_ci		No	Ninguna
<input type="checkbox"/>	14 URL_IMAGEN	varchar(150)	utf8mb4_spanish_ci		Sí	NULL
<input type="checkbox"/>	15 N_CLICKS	int			No	0

Figura 3.3: Base de Datos final

Se puede observar que existe una nueva columna llamada 'N_CLICKS' que no estaba en los datos originales, esta columna es la encargada de recoger el número de visitas que tiene el evento, de manera que podamos mostrarle al usuario en la pantalla principal los eventos más visitados.

3.1.3. Actualización automática

Una vez tenemos nuestra base de datos creada necesitamos introducirle los datos. Para introducir los datos usé Python y el paquete *'mysql-connector-python'*(6).

```
import mysql.connector

def update_bd():
    db = mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="----",
        database="-----",
        auth_plugin="mysql_native_password")

    mycursor = db.cursor()
    data = urlopen("https://datos.canarias.es/catalogos/general/dataset/11dc9d88-b456-4ae2-
    datafile = StringIO(data)
    csvReader = list(csv.reader(datafile))

    for row in csvReader[1:]:
        print(row)
        mycursor.execute("INSERT IGNORE INTO Eventos "
            "(FECHA_INICIO, FECHA_FIN, DURACION_DIAS, TIPO_EVENTO, ISLA, MUNICIPIO,
            "VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)

    db.commit()
    mycursor.close()
    db.close()
```

Figura 3.4: Código para insertar datos

Se puede observar el uso de **INSERT IGNORE** este comando nos ayuda a evitar introducir líneas que ya existan en nuestra base de datos, dando pie al siguiente concepto, conseguir una actualización automática que nuestro código aún no nos proporciona.

Para lograrlo necesitamos otro paquete de Python *'schedule'*(7). Este paquete nos permite planificar ejecuciones de código a lo largo del tiempo. En mi caso decidí actualizar la base de datos todos los días a las 00:00, para ello a continuación de mi código anterior introduje lo siguiente:

```
schedule.every().day.at("00:00").do(update_bd)

while True:
    schedule.run_pending()
    time.sleep(3600)
```

Figura 3.5: Código para actualización automática

Con esto ya tendría la base de datos actualizándose a diario sin necesidad de introducir los datos manualmente en caso de que hubiera alguna modificación.

3.1.4. XAMPP

XAMPP(11) es un entorno de desarrollo con PHP completamente gratuito y fácil de instalar. Lo más atractivo de este entorno es la facilidad de conectar nuestra base de datos en MySQL con el mundo exterior.

Una vez instalado y configurado (Prestando especial atención en la asignación de puertos, ya que, a veces los puertos por defectos están siendo usados por otros programas) al ejecutar el controlador nos aparecerá una pantalla como la siguiente:

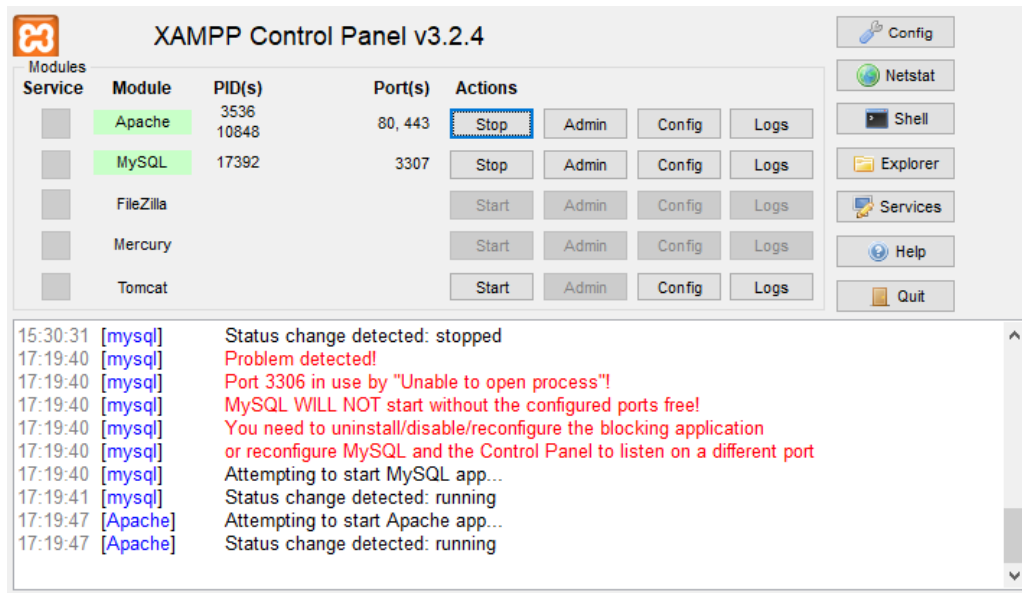


Figura 3.6: XAMPP panel de control

Indicando que nuestra base de datos ya está enlazada al servidor Apache. Para ver los datos y la información de nuestra base de datos debemos abrir un navegador e insertar nuestra IP seguida de la ruta /phpmyadmin/. En mi caso se me muestra de la siguiente forma:

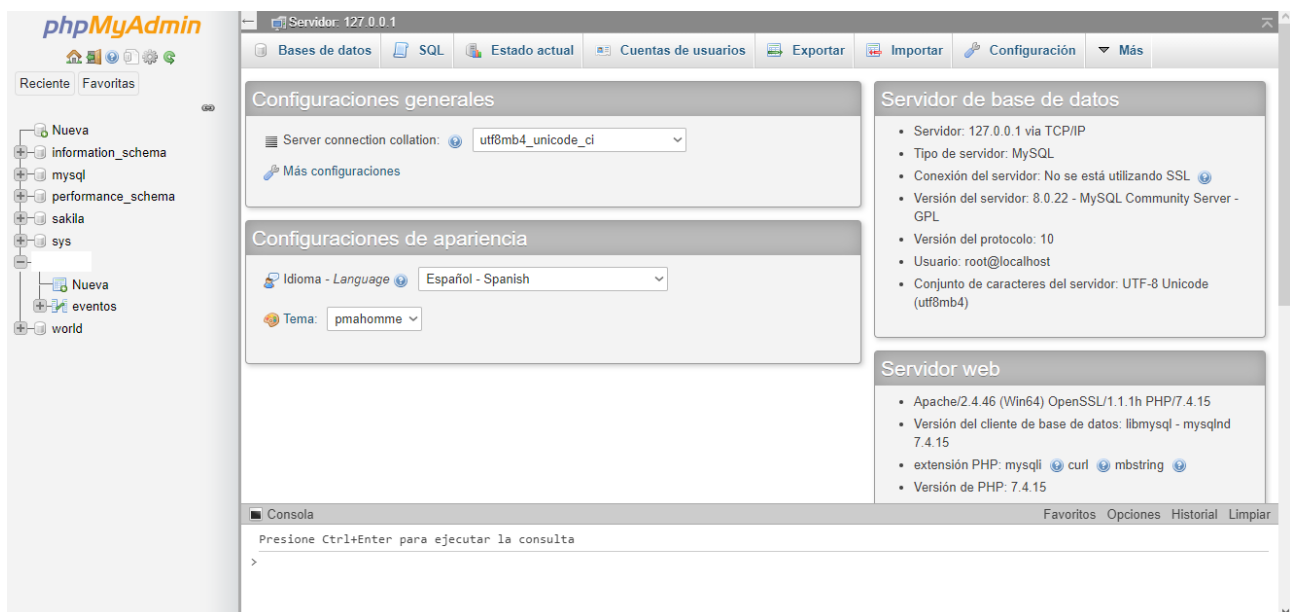


Figura 3.7: XAMPP panel de control Web

3.1.5. WebServices

Una vez tenemos todo conectado necesitamos crear los scripts en php para poder acceder a la información de la base de datos desde nuestro cliente. Para crear los script

necesitamos ir a la ruta local donde está instalado el xampp, una vez ahí dentro de la carpeta *htdocs* creamos un directorio, el cual, va a recoger nuestros scripts(webservices).

Conexión - **conexion.php**

Este script consiste en la creación de un enlace entre el php y la base de datos, en caso de error devolvemos un mensaje informativo. En los siguientes webservices veremos como debemos llamar primero a la conexión antes de poder llegar a la base de datos.

```
<?php
$hostname='localhost';
$database='---';
$username='---';
$password='---';

$connection=new mysqli($hostname,$username,$password,$database);
if($connection->connect_errno){
    echo "Lo sentimos, el sitio web está experimentando problemas";
}
?>
```

Figura 3.8: **conexion.php**

Buscar Eventos - **buscar_evento.php**

En este primer script más complejo empezamos a necesitar parsear la url para obtener los campos en los que está interesado el usuario. Un ejemplo de la entrada que podemos recibir sería la siguiente (suponemos que compone una url completa):

```
<IP>/mywebservices/buscar_evento.php?
startDate=14-4-2021&endDate=28-4-2021&eventType=Varios&
island=La %20Palma&municipality=muní&space=spa&place=place&
title=title&description=descdesc2&
```

Prácticamente dejamos que el usuario sea capaz de filtrar por cualquier valor. Dentro de los valores posibles los he dividido en 3 tipos:

- Fechas - [startDate, endDate]
- Seleccionables (Desde el frontend se obligará al usuario a elegir de un selector, al ser valores no variables) - [island, eventType]
- Cadenas - [municipality, space ,place ,title ,description]

```

<?php

include 'conexion.php';

$cond = array();
$query = "select DATE_FORMAT(FECHA_INICIO, '%d-%m-%Y'),
DATE_FORMAT(FECHA_FIN, '%d-%m-%Y'),
DURACION_DIAS,
TIPO_EVENTO,
ISLA,
MUNICIPIO,
DENOMINACION_ESPACIO,
LUGAR,
TITULO,
DESCRIPCION,
HORA,
MINUTO,
MAS_INFO,
URL_IMAGEN from eventos";

```

Figura 3.9: buscar_evento.php - Cabecera

Podemos observar como en la primera línea se hace referencia a la conexión que creamos previamente y desde donde ejecutaremos la query posteriormente. En **\$cond** almacenaremos todas las variables que obtengamos de la url. Por último la query con la información que queremos extraer a falta de aplicar los filtros.

```

//Dates
if (isset($_GET["startDate"])){
    $startDate=$_GET['startDate'];
    $cond[] = "( FECHA_INICIO >= STR_TO_DATE('$startDate', '%d-%m-%Y') OR FECHA_FIN >= STR_TO_DATE('$startDate', '%d-%m-%Y') )";
}
if (isset($_GET["endDate"])){
    $endDate=$_GET['endDate'];
    $cond[] = "( FECHA_INICIO <= STR_TO_DATE('$endDate', '%d-%m-%Y') OR FECHA_FIN <= STR_TO_DATE('$endDate', '%d-%m-%Y') )";
}
//Selects
if (isset($_GET["island"])){
    $island=$_GET['island'];
    if($island != 'Todas'){
        $cond[] = "ISLA = '$island'";
    }
}
if (isset($_GET["eventType"])){
    $eventType=$_GET['eventType'];
    if($eventType != 'Todos'){
        $cond[] = "TIPO_EVENTO = '$eventType'";
    }
}
//Strings like
if (isset($_GET["municipality"])){
    $municipality=$_GET['municipality'];
    $cond[] = "MUNICIPIO LIKE '%$municipality%'";
}
{...}

```

Figura 3.10: buscar_evento.php - Parsing

Lo siguiente es, formar las diferentes condiciones dependiendo de los tipos expuestos anteriormente, previamente se comprueba con **isset()** que el parámetro existe en la url. Las condiciones para las *fechas* son considerablemente más grandes, ya que, tienen en cuenta la posibilidad de pasar solo uno de los valores o pasar ambos. Los tipo *selector* deben coincidir perfectamente mientras que los tipo *cadena* es aceptada con que coincida en algún lugar del valor.

```
//Query creation
if (count($cond) > 0){
    $query .= " WHERE " . implode(" AND ", $cond);
}

$result = $connection -> query($query);

$events = array();

while($row=$result -> fetch_array()){
    $events[] = array_map('utf8_encode', $row);
}

echo json_encode($events);
$result -> close();

?>
```

Figura 3.11: buscar_evento.php - Final

Por último, una vez tenemos todas las condiciones creamos la query final sumándole la cláusula **WHERE** seguido de las condiciones creadas. Realizamos la query a la base de datos a través de la conexión creada, iteramos por todos los eventos, teniendo en cuenta la codificación como se mencionó en el '*Análisis de los Datos*' y se devuelve como un objeto json.

```
[{"0":"05-02-2021","DATE_FORMAT(FECHA_INICIO,'%d-%m-%Y')":"05-02-2021","1":"05-02-2021",
2021","2":"0","DURACION_DIAS":"0","3":"Varios","TIPO_EVENTO":"Varios","4":"Tenerife","ISLA":'
',"DENOMINACION_ESPACIO":'---',"7":"En el Facebook y Youtube del IEHCAN que figura en +Info'
era. Inconsciente c\u00c3\u00b3smico","TITULO":"'Visiones de una era. Inconsciente c\u00c3\u00b3smic
Zamarbide","DESCRIPCION":"'Muestra del artista Guillermo V\u00c3\u00a1lquez
Zamarbide","10":"19","HORA":"'19","11":"00","MINUTO":"'00","12":"'http://www.iehcan.com/agenda/
{"0":"03-02-2021","DATE_FORMAT(FECHA_INICIO,'%d-%m-%Y')":"03-02-2021","1":"28-02-2021",
2021","2":"25","DURACION_DIAS":"'25","3":"'Varios","TIPO_EVENTO":"'Varios","4":"'La Gomera","IS
',"DENOMINACION_ESPACIO":"'---',"7":"'Varios espacios del municipio","LUGAR":"'Varios espacios d
de Vallehermoso 2021: \u00c3\u0081frica","9":"'Talleres, cursos, exposiciones, actuaciones, etc. Consulta
exposiciones, actuaciones, etc. Consultar programaci\u00c3\u00b3n en M\u00c3\u00a1s Info.,"10":"'--',"I
',"12":"'https://cutt.ly/v8kfn7Lu","MAS_INFO":"'https://cutt.ly/v8kfn7Lu","13":"'http://www.gobiernod
{"0":"04-02-2021","DATE_FORMAT(FECHA_INICIO,'%d-%m-%Y')":"04-02-2021","1":"27-02-2021",
2021","2":"'23","DURACION_DIAS":"'23","3":"'Varios","TIPO_EVENTO":"'Varios","4":"'La Palma","ISL
',"DENOMINACION_ESPACIO":"'---',"7":"'Online + varios espacios del municipio","LUGAR":"'Online -
siendo","TITULO":"'Carnaval 2021: Somos Carnaval y lo seguiremos siendo","9":"'Actividades varias onli
M\u00c3\u00a1s Info.,"DESCRIPCION":"'Actividades varias online, cursos, charlas, multitud de exposici
"11":"'---" "MINUTO":"'---
```

Figura 3.12: Ejemplo de Salida - Web

Top Eventos - top_events.php

Este script es bastante parecido al buscador con la diferencia de que este no depende de un input, por lo tanto, será siempre la misma consulta.

```
include 'conexion.php';

$cond = array();
$query = "select DATE_FORMAT(FECHA_INICIO, '%d-%m-%Y'),
DATE_FORMAT(FECHA_FIN, '%d-%m-%Y'),
DURACION_DIAS,
TIPO_EVENTO,
ISLA,
MUNICIPIO,
DENOMINACION_ESPACIO,
LUGAR,
TITULO,
DESCRIPCION,
HORA,
MINUTO,
MAS_INFO,
URL_IMAGEN from eventos ORDER BY `N_CLICKS` DESC LIMIT 6";
```

Figura 3.13: top_events.php

Como podemos comprobar la principal diferencia está en la última línea de la query, donde le decimos que nos ordene las entradas por la columna `N_CLICKS` de forma descendente, ya que, nos interesan los eventos más visitados en este caso. Por último como en la aplicación sólo necesitamos mostrar los primeros 6 le ponemos un límite para ahorrarnos tráfico innecesario de datos.

El resto del script es igual que en la Figura 3.11, enviar la query y crear el json con el resultado para devolvérselo al usuario.

Evento pulsado - event_click.php

Para este script necesitamos conocer el registro a modificar. Sabemos que la **clave primaria** de nuestra base de datos es la url en la columna `MAS_INFO` la cual es única para cada registro.

```
<?php
include 'conexion.php';

if (isset($_GET["id"])){
    $id=$_GET['id'];
    $query = "UPDATE eventos SET N_CLICKS = N_CLICKS + 1 WHERE MAS_INFO = '$id'";
    $result = $connection -> query($query);
    $product = array();
    $product[] = $result;

    echo json_encode($product);
}
?>
```

Figura 3.14: event_click.php

Este script es bastante sencillo, ya que, no necesitamos parsear la información ni crear condicionales complejos. El registro con el campo MAS_INFO igual al pasado por parámetro incrementa la columna N_CLICKS en una unidad.

3.2. FrontEnd

Para el desarrollo de la aplicación se ha trabajado en **Android Studio**(8). Un IDE que permite de manera sencilla y cómoda desarrollar tu propia aplicación para Android usando el patrón de arquitectura de software **Modelo-Vista-Controlador**(9) de Java. Dispone de una vista previa cuando se están editando vistas y un emulador para poder hacer pruebas reales sobre la aplicación.

3.2.1. Página principal

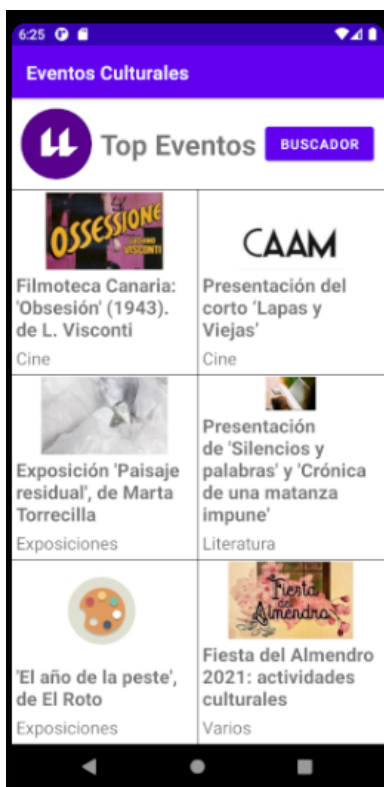


Figura 3.15: Página Principal

Al ejecutar la aplicación, se le muestra la pantalla 3.15 al usuario, en la cual, mostramos los 6 eventos más visitados. Si el usuario pulsa cualquiera de los eventos se le abren los detalles del mismo. En la parte superior mostramos la cabecera y el botón para redirigir al buscador.

Lo más complicado en la implementación ha sido utilizar los enlaces de imágenes para desplegarlas en la aplicación. Para resolverlo he usado la clase **NetworkImageView**, pese a ser bastante complicada a la hora de usarse para múltiples imágenes consiguió resolver el problema.

3.2.2. Buscador

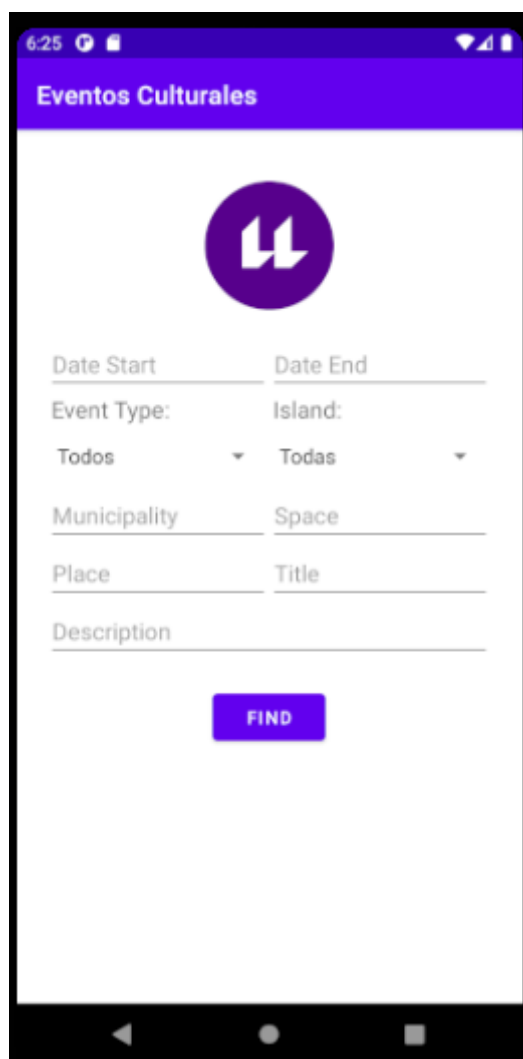


Figura 3.16: Buscador

En el buscador podemos observar los tres tipos de campos mencionados previamente (Fechas, seleccionables, cadenas). Al pulsar en una de las fechas se despliega un calendario para facilitar su uso al usuario. Los seleccionables muestran las opciones disponibles en la base de datos para los tipos de eventos y las islas, en caso de elegir la opción *Todos* no lo añadimos a los atributos de la url generada, ya que, la ausencia de un atributo significa que no se tiene en cuenta en la búsqueda. Por último, aparecen los campos de texto plano, actualmente necesita una coincidencia completa en la base de datos con el texto escrito.

Una vez se han introducido los datos (o ninguno en caso de querer consultar todos los eventos) se genera la url correspondiente como en el ejemplo de la subsección 3.1.5 y se realiza la petición al servidor. A continuación, se muestra el listado con los eventos coincidentes.

3.2.3. Lista de eventos



Figura 3.17: Lista de eventos

En el listado de eventos mostramos también la fecha de comienzo y finalización. Para el listado se ha usado **RecyclerView**, un tipo de vista en Android que a diferencia de la típica lista esta se crea bajo demanda. Esto quiere decir que no todas las filas se cargan en el momento del inicio, si no que, se cargan a medida que el usuario avanza por la lista, minimizando el tiempo de espera del usuario y dando la sensación de respuesta rápida.

El único inconveniente es la dificultad a la hora de programar los componentes necesarios para su funcionamiento, debido a la complejidad de las clases e interfaces. Esto conllevó a que la implementación del evento 'onclick' fuera más difícil de implementar de lo normal.

3.2.4. Detalles del evento

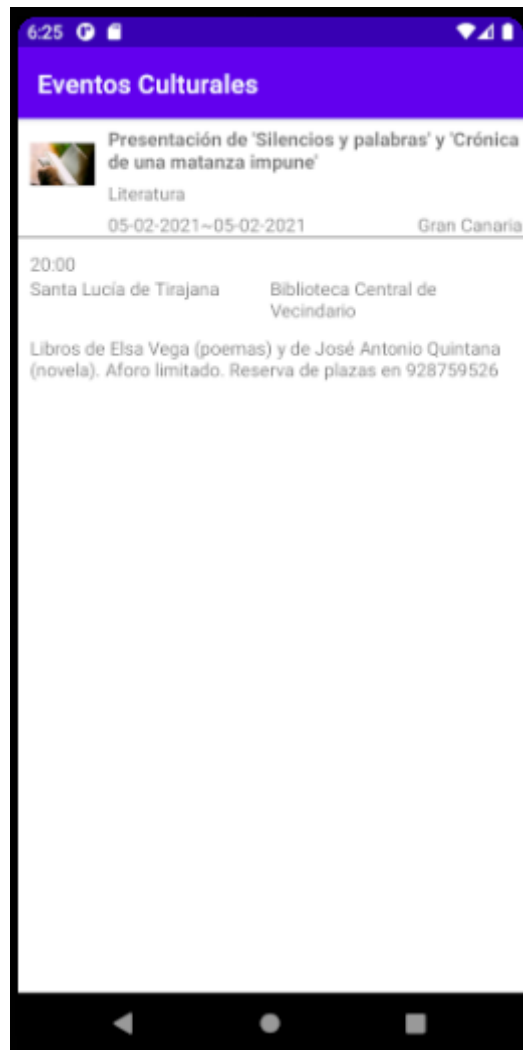


Figura 3.18: Detalles del evento

En la vista de los detalles del evento podemos ver como además de los campos mostrados en el listado aparecen la hora, lugar, municipio y la descripción del mismo.

3.2.5. Esquema final

Esquema genérico de la interacción entre los componentes. En el servidor, el script de python actualiza la base de datos una vez al día, extrayendo los datos de la fuente de información. El servicio de XAMPP está escuchando constantemente a través de los WebServices las peticiones realizadas por los usuarios a través de las aplicaciones instaladas en sus dispositivos móviles. Para satisfacer estas peticiones se conecta con la base de datos a través de un script en php y devuelve los datos requeridos al usuario.

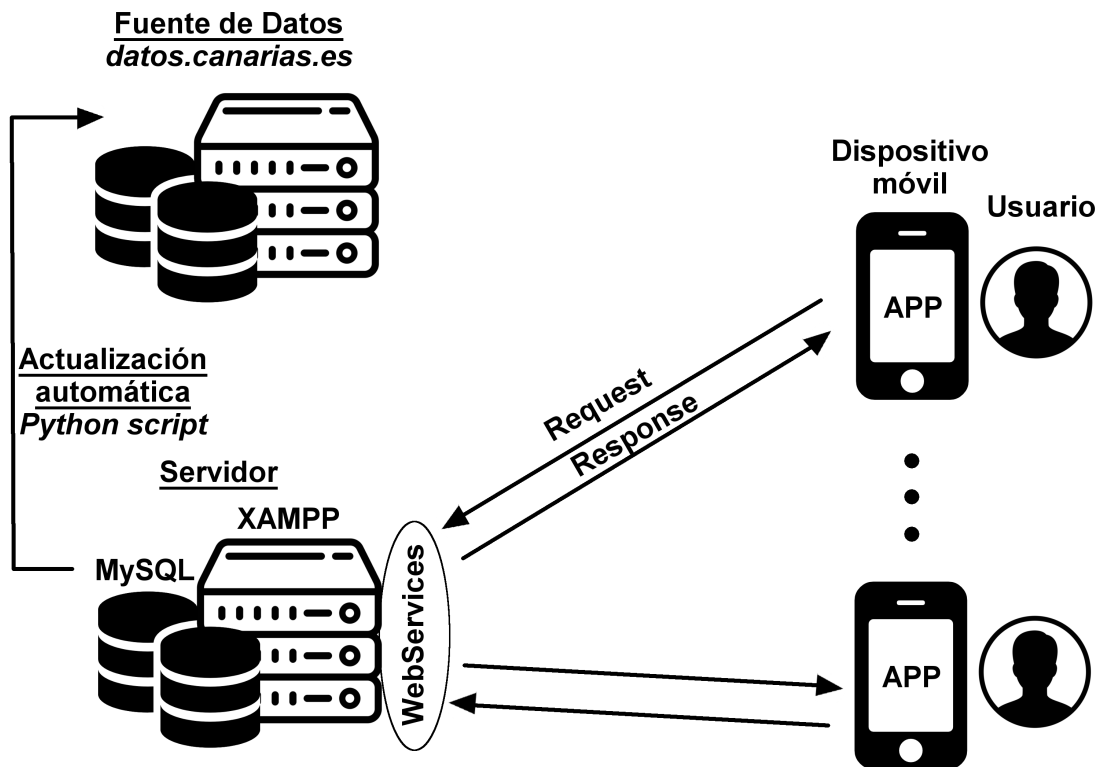


Figura 3.19: Esquema de los componentes

Capítulo 4

Despliegue

4.1. Modificaciones necesarias

Una vez todas las partes han sido presentadas, podríamos querer desplegar la aplicación para que cualquier usuario pueda beneficiarse de nuestros servicios.

A nivel de backend debemos montar la base de datos con MySQL y nuestro servidor Apache con XAMPP en una máquina que disponga de una IP pública (y los certificados correspondientes) para que nuestros usuarios sean capaces de alcanzar el servidor desde cualquier lugar de internet. Recordando que si queremos que se mantenga la base de datos actualizada debemos ejecutar el script 3.4 de Python en segundo plano.

A nivel del frontend (la aplicación Android) simplemente necesitamos cambiar la variable estática **MY_IP** por la IP pública donde tenemos nuestro servicio escuchando las peticiones y generar la APK desde la propia plataforma Android Studio, para posteriormente instalarla en cualquier dispositivo Android. En caso de querer subir la aplicación a la Play Store tiene una página web(10) con detalles a tener en cuenta y pasos a seguir.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

En conclusión, se ha desarrollado una aplicación que muestra al usuario los eventos culturales en canarias en base a sus necesidades y mostrando los eventos más visitados. Se han utilizado herramientas usadas por muchas aplicaciones y servicios.

Se ha realizado un gran trabajo de programación al haber implementado funcionalidades en 3 lenguajes diferentes (Python, PHP y Java) y haber conseguido enlazarlos todos en una única aplicación. Además del manejo de la base de datos e intentar buscar la mayor eficiencia tanto a nivel de servidor como en la aplicación Android.

En cuanto a lo personal, he adquirido muchos conocimientos en lo que respecta a crear una aplicación que requiere de un servidor y webservices para obtener los datos, además de la capacidad para poder adaptarme a los distintos lenguajes de programación.

5.2. Líneas Futuras

Existen variaciones y mejoras posibles sobre la aplicación:

- Login
- Manejo de reservas
- Añadir filtro post-búsqueda
- No mostrar en la pantalla principal los eventos ya terminados
- En la búsqueda separar las palabras de los campos de texto para hacerlo más permisivo
- Mejorar el diseño de las pantallas
- Obtener los datos de distintas fuentes y ponerlas en común
- Normalizar información de las tablas para facilitar la mejora anterior y tener la información más estructurada

Capítulo 6

Summary and Conclusions

6.1. Conclusions

In conclusion, I have developed an application that shows the user the cultural events in the Canary Islands based on their needs and showing the most visited events. Tools used by many applications and services have been used.

Hard programming work has been done by having implemented functionalities in 3 different languages (Python, PHP and Java) and having managed to link them all in a single application. In addition to managing the database and trying to find the highest efficiency both at the server level and in the Android application.

As for the personal, I have acquired a lot of knowledge about creating an application that requires a server and webservice to obtain the data, as well as the ability to adapt to different programming languages.

6.2. Summary

There are variations and possible improvements on the application:

- Login
- Reserve management
- Add post-search filter
- Do not show finished events on the main screen
- In the search, separate the words from the text fields to make it more permissive
- Improve screen design
- Get the data from different sources and put them in common
- Normalize information in the tables to facilitate the previous improvement and have the information more structured

Capítulo 7

Presupuesto

7.1. Desarrollo Software

El cálculo de la mano de obra se hace teniendo en cuenta los siguientes valores: Análisis (30€/h) y Programación(23€/h).

Desarrollo	Análisis (h)	Desarrollo (h)	Total (h)	Total (€)
Base de Datos	15	24	34	1002
WebServices	9	20	24	730
Aplicación Android	21	54	70	1.872
	45	98	143	3.602

Tabla 7.1: Resumen de Gastos - Desarrollo

7.2. Infraestructura Hardware

El hardware depende mucho del alcance que tenga la aplicación. Teniendo en cuenta que se limita a las Islas Canarias y responder consultas directas no se necesita una gran infraestructura. Actualmente existe la tendencia de alquilar servidores online, esto ayuda mucho en caso de no disponer del fondo monetario inicial o la infraestructura necesaria para mantener tu propio servidor. El principal problema de esta alternativa es la intervención y dependencia directa de otra empresa en nuestro servicio.

Pero actualmente lo más cómodo es recurrir al hosting con mensualidad y posibilidad de crecimiento dinámico. A continuación se muestra una comparación entre páginas que proporcionan este servicio.

	RAM	vCore	SSD	Coste (€/mes)
www.swhosting.com	4	2	60	12,20
clouding.io	4	2	60	16
es.kamatera.com	2	2	30	12

Tabla 7.2: Resumen de Gastos - Hosting

7.3. Total

	Coste Plano(€)	Coste Mensual(€)
Desarrollo Software	3.602	
Hosting		12,20

Tabla 7.3: Resumen de Gastos - Total

Bibliografía

- [1] Consejería de Educación, Universidades, Cultura y Deportes: Agenda Cultural,
<https://datos.canarias.es/catalogos/general/dataset/agenda-cultural>
- [2] TomaTicket,
<https://www.tomaticket.es>
- [3] Ticketea,
<https://www.ticketea.com>
- [4] Buscametas,
<https://www.buscametas.com>
- [5] MySQL,
<https://www.mysql.com>
- [6] Paquete de python para conexión con MySQL,
<https://pypi.org/project/mysql-connector-python/>
- [7] Paquete de python para planificar ejecuciones,
<https://pypi.org/project/schedule/>
- [8] Android Studio,
<https://developer.android.com/studio>
- [9] Modelo-vista-controlador,
<https://es.wikipedia.org/wiki/Modelo-vista-controlador>
- [10] Play Store - Subir APK,
<https://support.google.com/googleplay/android-developer/answer/9859152?hl=es>
- [11] XAMPP,
<https://www.apachefriends.org/es/index.html>