



DESARROLLO DE APLICACIÓN ANDROID: GUÍA TURÍSTICA PARA EL MUNICIPIO DE CULLERA.

Inmaculada Cárcel Puig

Tutor: Francisco José Martínez Zaldívar

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2018-19

Valencia, 3 de julio de 2019



Resumen

En la actualidad la sociedad se encuentra totalmente digitalizada, se utilizan dispositivos electrónicos y las aplicaciones que estos soportan para facilitar, en todos los ámbitos, el día a día de las personas. El presente Trabajo Fin de Grado consiste en la creación de una aplicación móvil desarrollada para sistemas operativos Android. Se trata de un asistente para facilitar el turismo en la ciudad de Cullera, exponiendo todo tipo de actividades para el disfrute del usuario dependiendo de la época del año en la que visite el municipio.

El desarrollo de la aplicación se ha llevado a cabo en diferentes fases. En primer lugar se ha diseñado una interfaz sencilla y atractiva para el usuario con el fin de agilizar su posterior programación, tras ello se ha pasado a la recopilación de información e imágenes necesarias y por último se ha realizado la implementación y desarrollo de la aplicación mediante Android Studio utilizando tecnologías como Java y XML, además de los Servicios Web de Google Maps.

Resum

Actualment la societat està totalment digitalitzada, s'utilitzen dispositius electrònics i les aplicacions que aquests suporten per a facilitar, en tots els àmbits, el dia a dia de les persones. El present Treball Fi de Grau consisteix en la creació d'una aplicació mòbil desenvolupada per a sistemes operatius Android. Es tracta d'un assistent per a facilitar el turisme a la ciutat de Cullera, exposant tota classe d'activitats per a garantir que l'usuari gaudisca, depenent de l'època de l'any en què visite el municipi.

El desenvolupament de l'aplicació s'ha dut a terme en diferents etapes. En primer lloc s'ha elaborat un primer disseny senzill i atractiu per a l'usuari amb la finalitat d'agilitzar la seua posterior programació, després s'ha passat a la recopilació d'informació i imatges necessàries i finalment s'ha realitzat la implementació i desenvolupament de l'aplicació mitjançant Android Studio utilitzant tecnologies com Java i XML, a més dels Serveis Web de Google Maps.

Abstract

Nowadays society is completely digitized, electronic devices and applications that they support are used to make easy, in all areas, people's routine. This Final Degree Project consists on the creation of a mobile application developed for Android operating systems. It is an assistant to facilitate tourism in the city of Cullera, exposing all kinds of activities for the enjoyment of users depending on when they visit the town.

The application development has been carried out in different phases. In the first place, a simple and attractive interface for the user has been designed in order to speed up its programming. Then, necessary information and images have been collected and finally the implementation and development of the application has been carried out through Android Studio using technologies such as Java and XML, in addition to Google Maps Web Services.



Índice general

1.	Introducción	4
2.	Objetivos	5
3.	Metodología de trabajo.....	6
3.1.	Gestión del proyecto.....	6
3.2.	Distribución en tareas.....	7
3.3.	Flujograma temporal	8
4.	Estudio de mercado.....	9
5.	Descripción del producto	10
5.1.	Funciones y requisitos de la aplicación.....	10
5.2.	Características de los usuarios.....	10
6.	Contexto tecnológico.....	11
6.1.	Windows	11
6.2.	Android Studio	11
6.3.	Android	12
6.4.	Servicios Web de Google Maps	12
6.5.	Firebase	12
6.6.	Java.....	13
6.7.	XML.....	13
7.	Estructura de la aplicación	14
7.1.	Entorno de desarrollo	14
7.2.	Activities	14
7.3.	Layouts.....	14
7.4.	Intents	15
7.5.	Fichero Manifest	15
7.6.	Servicios Web de Google Maps	15
7.7.	Base de datos: Firebase	16
8.	Desarrollo y resultados del trabajo.....	17
8.1.	Preparación previa a la programación	17
8.1.1.	Diseño previo a la programación.....	17
8.2.	Estructura de las activities.....	18
8.2.1.	Archivo Strings	18
8.2.2.	Carpeta Drawable.....	19
8.2.3.	Cambio de letra y sombreado.....	19
8.2.4.	Activities	20



8.2.4.1.	MainActivity	20
8.2.4.2.	Contactos.....	21
8.2.4.3.	Menús	23
8.2.4.4.	Activities informativas	25
8.2.4.5.	Mapas	27
8.2.4.6.	SitiosWeb	29
8.3.	Funcionamiento de la aplicación: Flujogramas y explicación	29
8.3.1.	Flujograma principal	30
8.3.2.	Flujograma actividades culturales	32
8.3.3.	Flujograma alojamiento.....	33
8.3.4.	Flujograma aparcamiento	34
8.3.5.	Flujograma deporte y tiempo libre	35
8.3.5.1.	Flujograma deportes de equipo	36
8.3.5.2.	Flujograma rutas en bici	37
8.3.5.3.	Flujograma senderismo	38
8.3.5.4.	Flujograma actividades acuáticas	39
8.3.6.	Flujograma gastronomía.....	40
8.3.7.	Flujograma ocio.....	41
8.3.7.1.	Flujograma ocio invierno	42
8.3.7.2.	Flujograma ocio verano.....	43
8.3.7.3.	Flujograma Pascua	43
8.3.7.4.	Flujograma sanidad	44
8.3.8.	Flujograma supermercados.....	44
8.3.9.	Flujograma talleres	45
8.3.10.	Flujograma transporte	45
8.3.11.	Flujograma vacaciones en familia	46
9.	Conclusiones y propuesta de trabajo futuro	47
9.1.	Conclusiones	47
9.2.	Propuesta de trabajo futuro	47
10.	Bibliografía	48



Índice ilustraciones

Ilustración 1. Pantalla Android Studio.	11
Ilustración 2. Pantalla Firebase.	12
Ilustración 3. Ejemplo de estadística aportada por Firebase: tiempo medio que utilizan los usuarios en la aplicación y pantallas principales.	16
Ilustración 4. Ejemplo archivo Strings.	18
Ilustración 5. Ejemplo del cambio de letra con sombreado.	20
Ilustración 6. Pantalla de inicio de la aplicación.	21
Ilustración 7. Pantalla contactos.	22
Ilustración 8. Ejemplo de menú.	24
Ilustración 9. Ejemplos del texto seleccionable.	26
Ilustración 10. Ejemplos de mapas.	28



1. Introducción

La realización del presente Trabajo de Fin de Grado (TFG) posibilita la finalización de mis estudios del Doble Grado en Administración y Dirección de Empresas + Ingeniería de Tecnologías y Servicios de Telecomunicación, y de esta manera permite la aplicación a un proyecto real de los conocimientos adquiridos a lo largo de la carrera y la adquisición de nuevos conocimientos de forma autónoma.

Tras haber cursado la mayoría de las asignaturas que componen la titulación, decidí dedicar mi Trabajo Final de Grado a la programación de una aplicación móvil.

Algunas de las asignaturas que me permitieron familiarizarme con la programación fueron: Programación, Fundamentos de sistemas digitales, Sistemas digitales programables, Sistemas microprocesadores, Diseño de servicios Telemáticos, así como todas aquellas en las que hacíamos uso de Matlab para la solución de problemas en las clases prácticas.

La elección de llevar a cabo una aplicación móvil se debe a que la gran mayoría de nuestro tiempo lo pasamos utilizando nuestro Smartphone, y por tanto, las apps que este soporta, la media española de tiempo utilizado en apps son 7,7 horas semanales. Además, solo en España se descargan al día 4 millones de aplicaciones y el aspecto más valorado de estas, en un 72%, es que cubran una necesidad [1]. Por lo que me parecía interesante que el fruto final de mi trabajo fuese algo útil y que la gente pudiese utilizar de una manera cómoda.

En la actualidad las aplicaciones se utilizan en todos los ámbitos, tanto laborales como personales, por lo que hay muchos tipos de aplicaciones diferentes. La elección del tipo de aplicación que he llevado a cabo estuvo influida por varios factores.

En primer lugar, la dificultad que encuentran en muchas ocasiones los turistas de encontrar actividades e información sobre el destino de su viaje de manera sencilla y rápida. Por lo que sería una herramienta útil una guía que aportara información sobre todas las actividades y atractivos que presenta un destino turístico para sus visitantes, accediendo a esta herramienta con tan solo unos clics desde el dispositivo más utilizado en nuestro día a día: el smartphone.

Por otro lado, la ciudad escogida como objeto de mi aplicación fue Cullera debido al conocimiento que tengo sobre ella puesto que es mi lugar de residencia, así como también por mi experiencia laboral en empresas del sector turístico en la localidad.

De esta manera surgió mi idea de TFG: Una aplicación a modo de guía que facilitara la visita y la estancia a los turistas de Cullera a la que titulé Disfruta Cullera, título que ejemplificaba aquello que se quería conseguir mediante la aplicación.



2. Objetivos

El principal objetivo de este Trabajo Final de Grado es conseguir desarrollar una aplicación móvil que facilite y mejore la experiencia del turismo en la ciudad de Cullera.

Un desarrollo de la interfaz de una aplicación atractivo y sencillo para el usuario es uno de los aspectos más importantes para el triunfo de esta [2].

Por ello, realicé un primer trabajo de investigación que consistió en encontrar aplicaciones que estuvieran relacionadas con el turismo en alguna localidad, y así lograr encontrar aspectos que podían ser positivos para la aplicación, descartar aquellos que no y añadir nuevos que no habían sido contemplados en otras aplicaciones y podían ser de utilidad para los usuarios.

Un aspecto interrelacionado con el diseño y muy importante en el triunfo de una app es la usabilidad [3]. La usabilidad engloba la sencillez y la calidad de navegación de la app: si no hay errores o bloqueos, si el usuario encuentra los contenidos que quiere...

Por lo que el objetivo era conseguir una aplicación intuitiva y fácil de manejar, de forma que el usuario accediera a la información que necesitase rápidamente.

Otro aspecto a tener en cuenta es la utilidad de la aplicación, qué motivos harán al usuario utilizarla en vez de utilizar otros canales para los mismos fines. Aquí entra la finalidad misma de Disfruta Cullera, tener al alcance toda la información necesaria y que puede resultar útil en un viaje en un mismo entorno, sin tener que acceder a diferentes fuentes.

Por otro lado, también he implementado una base de datos con la finalidad de recopilar información sobre el uso de la aplicación, ya que esta podría usarse por parte de alguna empresa o institución y sería una manera de encontrar información sobre las actividades o lugares más interesantes para los turistas.

3. Metodología de trabajo

3.1. Gestión del proyecto

Para la realización de dicho Trabajo Final de Grado establecí una serie de tareas previas al desarrollo de la aplicación en sí misma, otras para su programación y otras posteriores a la finalización de la programación. Así, a continuación, realizaré una breve exposición de las mismas, explicando el proceso seguido de forma rápida.

Para empezar, escogí qué tipo de sistema operativo quería utilizar. Decidí escoger los sistemas Android debido a que más del 80% de los dispositivos a nivel mundial utilizan este sistema operativo [4]. Además mis dispositivos también son Android, cosa que facilitaría mi trabajo.

Tras ello, necesitaba familiarizarme con el entorno de desarrollo correspondiente a Android. Para ello, escogí realizar un curso de Android Studio para principiantes mediante un tutorial de YouTube [5]. Por otro lado, debido a la baja capacidad de mi ordenador, no era posible lanzar un simulador virtual para la app en este, así que llevé a cabo la activación de las opciones de desarrollador de mi dispositivo móvil para poder probar en este los ejercicios del tutorial y las versiones de mi aplicación que iba llevando a cabo.

El seguimiento del tutorial fue de ayuda en diferentes aspectos: conocer más el entorno de desarrollo Android Studio, aprender a instalar la app en el Smartphone para las pruebas, realizar un recordatorio del lenguaje Java y conseguir la familiarización con XML.

Tras esto, empezó lo que conocemos como el ciclo de vida de desarrollo software.¹

Para empezar, en la etapa de análisis de requisitos, busqué aplicaciones ya existentes sobre turismo. De esta manera, me inspiré sobre cómo podría ser Disfruta Cullera, qué opciones podría implementar de las observadas y cuáles nuevas podría añadir. Después pasé a recopilar toda la información necesaria para la aplicación, tanto imágenes como información, datos de contacto, etc. Una vez aquí, realicé un listado de las alternativas definitivas que ofrecería la aplicación.

En la siguiente fase, fase de diseño, decidí hacer un diseño inicial en papel. Para ello, detallé la estructura que seguiría la aplicación y cómo estarían interrelacionadas las *activities* que la compondrían.

De esta manera, llegué a la fase de codificación, en esta me dispuse a plasmar mis ideas mediante Android Studio. Tras la creación del proyecto, empecé una programación gradual, implementando pantallas por niveles, primero los menús principales, los secundarios y finalmente las pantallas informativas. Todo ello mientras me apoyaba en la ayuda de mi tutor y ayudas de Internet e iba redactando el proceso que seguía para poder detallarlo posteriormente en esta memoria.

Finalizada la programación de mi primer prototipo, pasé a la fase de pruebas que se dividió en dos subfases: en la primera me aseguré de que el prototipo elaborado no tenía errores de funcionamiento, la segunda consistió en dejar probar la aplicación a gente que no conocía la localidad en la que se centraba esta. La elección de estas personas tuvo la finalidad de cubrir todos los tipos de turistas que pudieran visitar la localidad.

¹ Ciclo de vida de desarrollo software: secuencia estructurada y definida de las etapas para desarrollar un producto software deseado.



Esto me llevó a la siguiente fase, correcciones e implementación. En esta fase solventé ciertos errores que había cometido en el prototipo inicial, modifiqué aspectos de la aplicación que no eran atractivos y añadí ideas nuevas.

Así llegué a la fase de verificación, que consistió en instalar la aplicación en el dispositivo móvil y realizar diversas pruebas con el fin de comprobar que no existían errores en la misma.

Una vez conseguí el diseño final sin errores, pasé a la redacción de la memoria que recogería la información sobre todo el proceso que se había seguido, con la ayuda de los apuntes tomados durante la elaboración del proyecto.

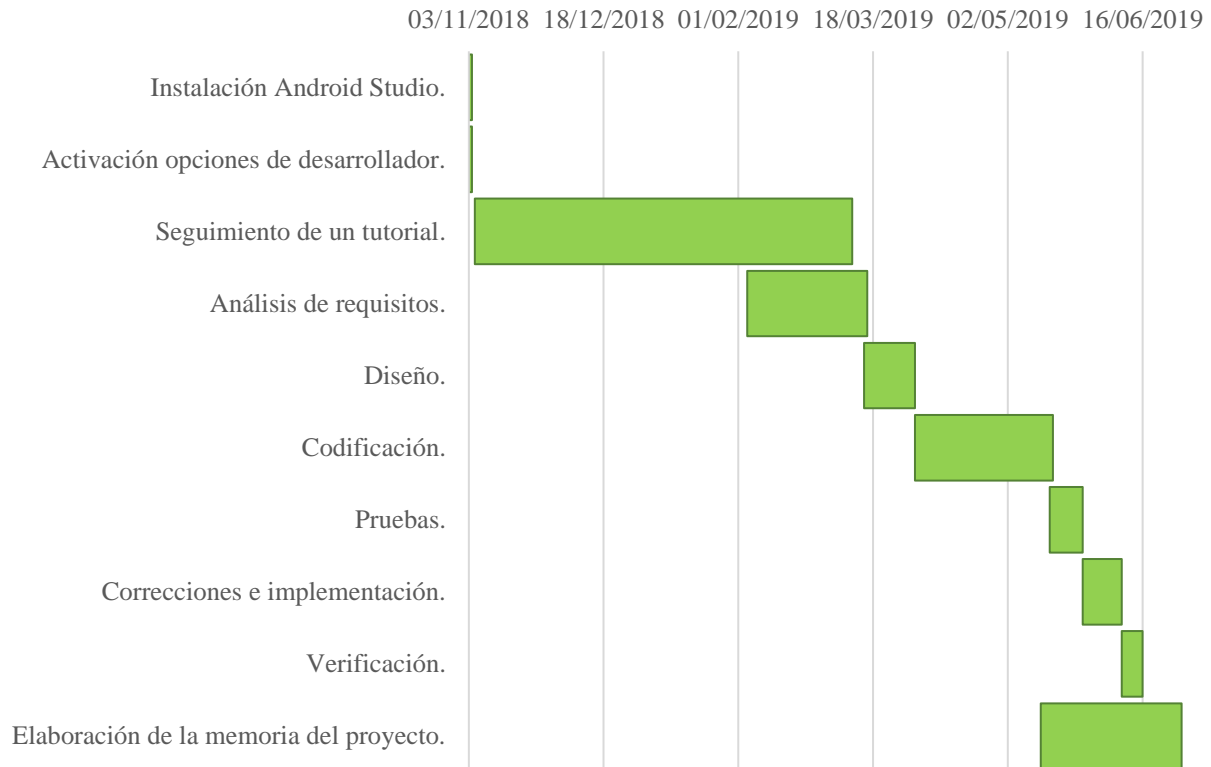
3.2. Distribución en tareas

A continuación redactaré una lista de tareas seguidas para llevar a cabo este proyecto. Entre estas podemos observar el ciclo de vida de desarrollo software y sus fases.

- Instalación de entorno de desarrollo Android Studio.
- Activación de las opciones de desarrollador en el Smartphone para posibilitar posteriores instalaciones de la aplicación en el mismo.
- Seguimiento de un tutorial para la familiarización con el entorno de desarrollo.
- Ciclo de software:
 - o Análisis de requisitos: Búsqueda y estudio de apps turísticas existentes, recopilación de la información necesaria para el desarrollo del proyecto e ideación de las funciones de la app.
 - o Diseño: Elaboración de un diseño inicial en papel en el que se detalla la estructura que seguirá la app respecto a las relaciones de sus *activities*.
 - o Codificación: Implementación del código necesario para desarrollar el prototipo inicial de la aplicación.
 - o Pruebas: Comprobación del correcto funcionamiento del prototipo diseñado y prueba del mismo con diferentes tipos de posibles turistas.
 - o Correcciones e implementación: Aplicación de las correcciones necesarias e implementación del diseño final.
 - o Verificación: Comprobación de la inexistencia de fallos en la aplicación final.
- Elaboración de la memoria del proyecto.

3.3. Flujograma temporal

Para una vista más clara del modo de elaboración del proyecto presento a continuación un gráfico que representa un flujograma temporal que contiene las tareas anteriormente citadas.



Gráfica 1. Flujograma temporal de tareas.

4. Estudio de mercado.

En este apartado detallaré el proceso seguido para construir la idea de mi aplicación mediante la búsqueda y consulta de apps de la misma temática.

Para la búsqueda de aplicaciones con fines turísticos utilicé la plataforma Google Play y descargué diversas aplicaciones: *Benidorm* [6], *CulleraTurismo* [7], *Gandia* [8] y *Xàbia* [9]. La elección de las aplicaciones fue motivada por similitudes de estas ciudades con la que iba a tratar en mi aplicación: ciudades de costa con turismo mayoritariamente de sol y playa. Tras analizar el funcionamiento de estas aplicaciones y las opciones que ofrecían, escogí algunas de estas como útiles para la aplicación y descarté otras que no me parecían atractivas.

Cullera ya disponía de una aplicación administrada por el ayuntamiento, del estilo de la que yo iba a elaborar aunque de carácter profesional, desarrollada por la empresa 360 Movic². Además, la elaborada a lo largo del trabajo incluiría aspectos que no aparecían en la ya existente y la estructura de ambas sería distinta. Un aspecto que me gustó de la aplicación utilizada por el ayuntamiento de Cullera fue el diseño de los botones, en los que se utilizaban imágenes como fondo de los mismos, promocionando así el municipio en todas las pantallas de la aplicación. Sin embargo, me parecía una excesiva carga de información las largas explicaciones que se realizaban en algunas pantallas. También presentaba una estructura poco práctica, que hacía un tanto complicado el acceso a algunos sectores de la información proporcionada.

La aplicación que correspondía a la ciudad de Benidorm pertenece a un grupo de aplicaciones que forma una red social de viajeros denominada Minube³. No me convencía la estructura que presentaba la aplicación, por lo que de esta aplicación solo analicé los contenidos que ofrecía.

Respecto a la aplicación correspondiente a Gandía, no me resultó atractivo el diseño, pues no aprovechaba imágenes de la ciudad para promocionarla y hacía uso en todas las pantallas de un fondo único. Sin embargo, tenía unos contenidos muy completos.

Por último, la aplicación de turismo en Jávea, desarrollada por la empresa Idea Mixta⁴, tenía gran variedad de contenidos, a pesar de que el diseño no era el ideal para mi gusto. Esta aplicación me aportó la idea para la estructura de mi aplicación, pues dividía el acceso a las fiestas en meses, aspecto que yo he utilizado como estructura principal de mi aplicación.

De esta manera, el proceso seguido de consulta de las aplicaciones me ayudó a escoger qué estructura seguiría mi aplicación, me dio ideas sobre el diseño a elaborar y sobre gran variedad de información que incluir en la aplicación.

Un aspecto que no encontré en ninguna de las aplicaciones y que me pareció interesante añadir en mi proyecto fue la sección de aparcamiento. Así se proporciona al usuario información sobre en qué zonas es más fácil aparcar y en cuáles es gratuito o no. Consiguiendo esto aportar valor añadido a mi Disfruta Cullera.

² 360 Movic: Empresa formada por un equipo de profesionales especialistas en la digitalización de PYMES que realizan desarrollo de aplicaciones móviles .[10]

³ Minube: Portal o red social de viajeros que desarrolla aplicaciones utilizadas como guías de viaje. [11]

⁴ Idea Mixta: Empresa que desarrolla soluciones innovadoras para todo tipo de empresas, entre ellas el diseño de aplicaciones móviles. [12]

5. Descripción del producto

A continuación expondré los detalles de la aplicación elaborada a lo largo del proyecto, Disfruta Cullera. Para ello explicaré para qué funciones ha sido diseñada y qué requisitos debe cumplir para ello, así como cuál es su público objetivo.

5.1. Funciones y requisitos de la aplicación

La aplicación desarrollada está elaborada para que funcione como una guía turística a disposición de los usuarios. De esta manera, estas serán sus principales funciones:

- Proporcionar datos de contacto de instituciones de interés para los turistas.
- Dar información acerca de todas las opciones que ofrece el municipio: alojamiento, gastronomía, ocio, deporte, tiempo libre, cultura, sanidad, supermercados, transporte, talleres...
- Aportar toda la información necesaria sobre cada una de estas opciones.
- Ofrecer accesos desde la aplicación para ampliar la información proporcionada.
- Facilitar el acceso a la información por parte de los usuarios.
- Recopilar información sobre las interacciones del usuario.

Para cumplir cada una de estas funciones y conseguir la satisfacción de los usuarios la aplicación deberá cumplir una serie de requisitos:

- Presentar una estructura sencilla.
- Distribuir la información de una forma simple para el usuario.
- Presentar datos de contacto correctos.
- Aportar accesos a páginas útiles para el usuario.
- No presentar retardos ni fallos en el funcionamiento.

5.2. Características de los usuarios

La aplicación desarrollada se dirige a cualquier persona que visite la ciudad de Cullera. Sin embargo, al tratarse de una aplicación móvil desarrollada para sistemas operativos Android, el público objetivo será toda persona que visite la ciudad de Cullera y que disponga de un dispositivo Android.

Además se dirige a personas de más de 18 años, pues son las que visitarán Cullera de manera autónoma. No hay un máximo de edad para el uso de la aplicación, pero a partir de los 65 años el uso de smartphones empieza a bajar. [13]

6. Contexto tecnológico

En este apartado se describirán las herramientas, lenguajes y entornos de programación utilizados para el desarrollo de la aplicación, sin hablar de detalles sobre el desarrollo, ya que estos serán abordados más adelante.

6.1. Windows

El sistema operativo del equipo utilizado para el desarrollo de la aplicación es Windows 10. Este cumplía los requisitos para soportar Android Studio pero no para emular dispositivos Android debido al tipo de procesador, por lo que, como he dicho anteriormente, las pruebas de la aplicación las realicé en mi dispositivo móvil.

6.2. Android Studio

El entorno de desarrollo integrado (IDE) oficial para aplicaciones Android se denomina Android Studio y está basado en IntelliJ IDEA⁵. [14]

Android Studio tiene muchísimas herramientas que pone a disposición de los usuarios, algunas de ellas son:

- Ayudas para traducción y consejos de optimización.
- Sistema de compilación basado en Gradle flexible.
- Entorno unificado que permite desarrollos para todos los dispositivos Android.
- Emulador rápido con distintas funciones.
- Instant Run que permite aplicar cambios mientras se ejecuta la app.

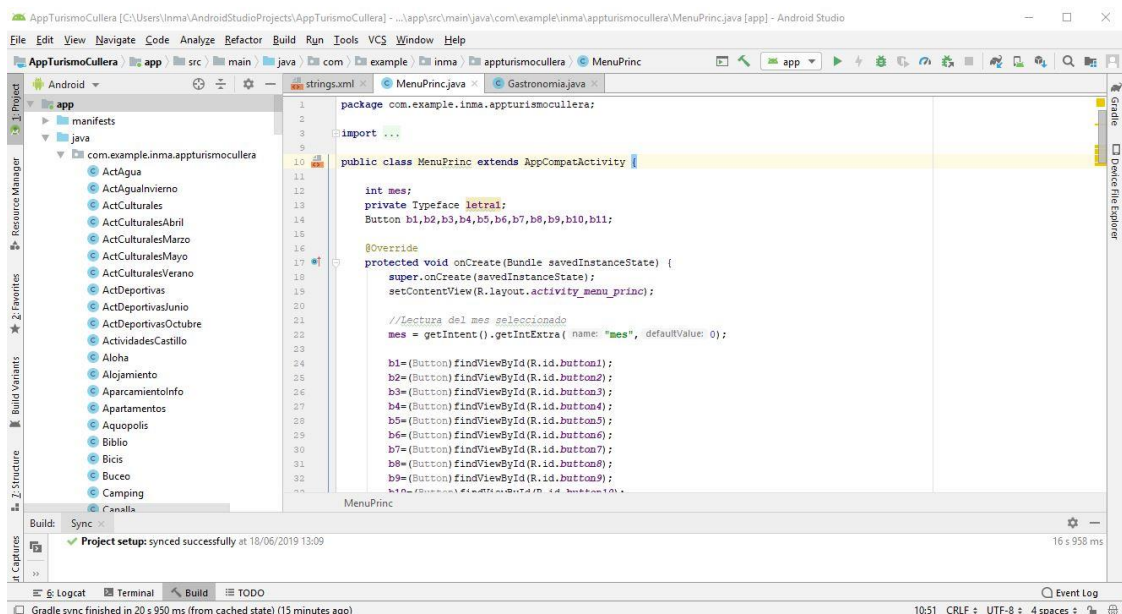


Ilustración 1. Pantalla Android Studio.

⁵ IntelliJ IDEA: Entorno de desarrollo integrado para desarrollar programas informáticos. Desarrollado por JetBrains y no basado en Eclipse.

6.3. Android

Android es un sistema operativo diseñado para dispositivos móviles táctiles como smartphones, tablets, relojes inteligentes, automóviles o televisores. Fue desarrollado por Google, y está basado en software de código abierto como en el Kernel de Linux. [15]

6.4. Servicios Web de Google Maps

En la aplicación desarrollada se han añadido mapas haciendo uso de la API de Google Maps. Gracias a esta se obtienen gran cantidad de funcionalidades, en concreto, para Disfruta Cullera, se han podido añadir markers, líneas y polígonos para indicar al usuario lugares o zonas.

6.5. Firebase

Plataforma para el desarrollo de aplicaciones web y aplicaciones móviles, propiedad de Google, aunque sus desarrolladores fueron James Tamplin y Andrew Lee. Está ubicada en la nube y crea proyectos sin necesidad de un servidor ya que las herramientas se incluyen en los SDK. [16] En el caso de la aplicación desarrollada, esta base de datos se utilizará para analizar las visitas a la aplicación mediante la herramienta de estadísticas: número de usuarios, información más solicitada, tiempo usado en la aplicación...

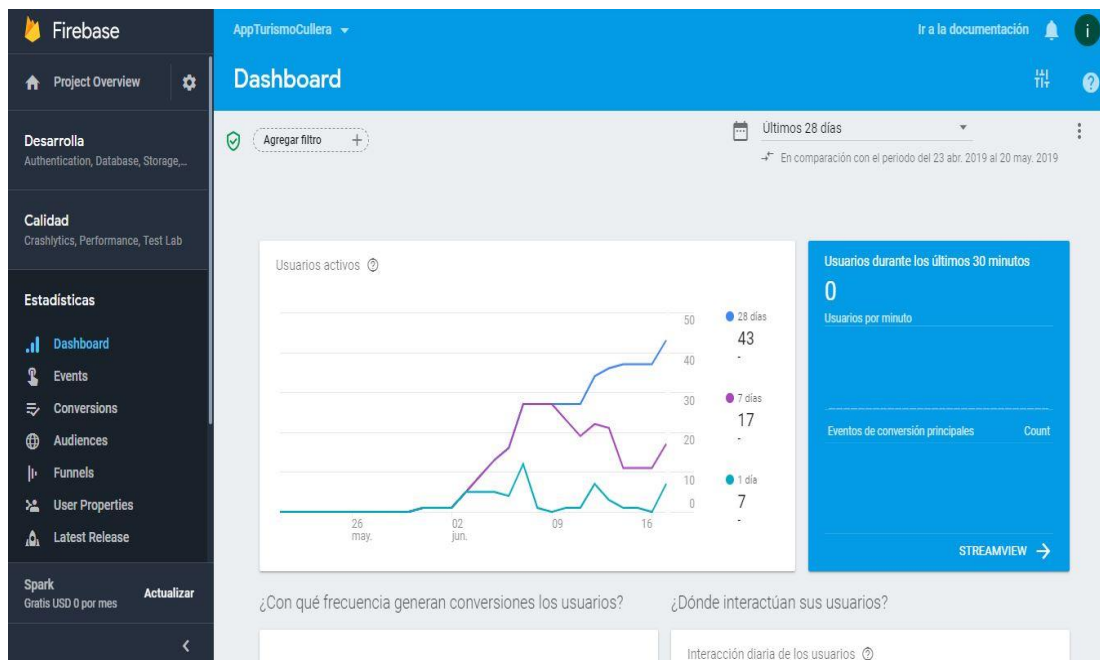


Ilustración 2. Pantalla Firebase.



6.6. Java

Java es el lenguaje de programación orientado a objetos más demandado actualmente. El principal objetivo de este lenguaje es que, una vez se cree el programa, pueda ser ejecutado en cualquier plataforma. Una ventaja de la utilización de Java en Android Studio es que Google proporciona muchas herramientas que sirven de ayuda. Además, Java es un lenguaje sencillo que permite crear programas complejos haciendo uso de órdenes básicas. [17]

6.7. XML

XML (eXtensible Markup Language) es un lenguaje basado en etiquetas. Mediante XML se puede almacenar información y datos legibles tanto para ordenadores como para personas. Mediante este lenguaje se crean y organizan los elementos que aparecerán en la pantalla de los dispositivos. [17]

7. Estructura de la aplicación

En este apartado se describirá el software utilizado para desarrollar la aplicación llevada a cabo en el presente Trabajo Final de Grado: Disfruta Cullera.

Disfruta Cullera tiene una arquitectura bidireccional. En un sentido, el usuario obtiene información de la aplicación. En el otro sentido, la base de datos vinculada a la aplicación recoge datos gracias a la recopilación de las interacciones del usuario, obteniendo información sobre las preferencias del mismo.

Por otro lado, la razón para escoger Java como lenguaje de programación es que es el lenguaje nativo Android. Además es un lenguaje con el que estaba familiarizada previamente a empezar este trabajo, ya que lo había utilizado en algunas de las asignaturas del grado.

7.1. Entorno de desarrollo

Una vez escogí el sistema operativo en el que funcionaría mi aplicación debía escoger el entorno de desarrollo para llevarla a cabo. Este fue Android Studio debido a que es el entorno de desarrollo integrado oficial de Android. [14]

Además el tutorial que escogí utilizaba este IDE.

7.2. Activities

Una *activity* es un ente básico de una aplicación, se puede decir que cada pantalla es una *activity*. Estas se componen de parte lógica y parte gráfica, y representan cada una de las pantallas de las apps. La parte lógica está compuesta por un archivo .java utilizado para interactuar, manipular y colocar el código de la *activity*, la parte gráfica corresponde a los *layouts*. [18]

La aplicación llevada a cabo está compuesta por muchas *activities* cuya relación y construcción se explicará más adelante.

Dentro de las *activities* se establecen distintos métodos para llevar a cabo las funciones que se quieren implementar.

7.3. Layouts

Los *layouts* son estructuras que marcan como se reparten los elementos en las diferentes pantallas. Estos son modificados mediante la edición de ficheros XML. De esta manera los *layouts* definen la parte gráfica de las pantallas de las aplicaciones.

Estos pueden ser visualizados mediante Android Studio durante su implementación. Existen diferentes tipos de *layouts*. Los utilizados para la implementación de esta aplicación son:

- `ConstraintLayout`: Este tipo de *layout* es muy flexible y fácil de usar desde el editor visual de Android Studio. En él se definen *constraints* (restricciones) entre los mismos elementos o con los extremos de la pantalla para fijar las posiciones. [19]

- `LinearLayout`: Aporta una gran comodidad para añadir widgets al proyecto. Estos permiten orientación vertical y horizontal, en la aplicación en cuestión se ha utilizado la vertical. Así, se pueden añadir los widgets uno tras otro, dejando una estructura ordenada de forma rápida. [20]

7.4. Intents

Los objetos de tipo `Intent` son utilizados para cambiar de una *activity* a otra y permiten enviar datos entre ellas. Existen objetos de tipo `Intent` explícitos e implícitos.

7.5. Fichero Manifest

Este fichero XML se incluye en todos los proyectos Android. En él se declaran todas las *activities* que componen la aplicación, además es necesario incluir los permisos de acceso software y hardware que sean necesarios para el funcionamiento de la aplicación. [21]

En el caso de la aplicación que he desarrollado, se han tenido que incluir los permisos para acceder a aplicaciones y sitios web mediante *WebViews*, y para utilizar las librerías de Google Maps.

Los permisos necesarios para acceder a sitios web, mediante *WebViews*, tanto calificados como seguros como no seguros han sido:

- `android.permission.INTERNET`: solicita acceso a Internet.
- `android.permission.ACCESS_NETWORK_STATE`: verifica el estado de Internet.

Para utilizar las librerías de Google Maps es necesario incluir el permiso `android.permission.ACCESS_FINE_LOCATION`.

7.6. Servicios Web de Google Maps

Debido a que en la aplicación se utilizan servicios de Google Maps es necesaria la creación de una clave para el acceso a los servidores donde se encuentran estos servicios. Para ello se ha seguido un tutorial sobre cómo obtener esta clave [22] y sobre cómo utilizar las herramientas para incluir en los mapas *markers*, polígonos y líneas.

Tras la obtención de dicha clave se incluyó en el archivo `AndroidManifest.xml` con el fin de que la aplicación obtuviera los permisos para poder utilizar mapas de Google Maps.

7.7. Base de datos: Firebase

En esta aplicación se ha hecho uso de la base de datos de Google Firebase. Para su implementación también se hizo uso de un tutorial. [23]

De esta manera integré la base de datos de google con mi aplicación Android Studio, permitiéndome esto más tarde acceder a estadísticas sobre el uso de la app.

Estas estadísticas aportan información muy útil:

- Número de usuarios que han utilizado la aplicación.
- Usuarios activos en distintos intervalos de tiempo.
- En qué pantallas interactúan más los usuarios.
- Cuánto tiempo utilizan en la aplicación y en cada una de las pantallas.
- Los fallos que tiene la aplicación.
- Cuántas sesiones de cada país se dan en la aplicación.
- Plataformas que utiliza la aplicación (en este caso solo hay una, Android)

Todos estos datos resultan de gran utilidad para un administrador puesto que puede mejorar la aplicación o potenciarla según los gustos de sus clientes.



Ilustración 3. Ejemplo de estadística aportada por Firebase: tiempo medio que utilizan los usuarios en la aplicación y pantallas principales.

8. Desarrollo y resultados del trabajo

Este apartado consistirá en la explicación detallada de todo el proceso seguido para llegar a desarrollar la aplicación.

8.1. Preparación previa a la programación

El primer paso seguido fue la instalación del software necesario para llevar a cabo el proyecto. Se instaló el JDK [24] necesario para poder realizar la programación y el IDE Android Studio [25].

Por otro lado, fue necesaria la activación de las opciones de desarrollador de mi Smartphone que permitirían instalar las aplicaciones en este más adelante, ya que, como he comentado anteriormente mi ordenador no permitía la emulación de dispositivos virtuales.

Tras ello realicé el seguimiento del tutorial. Este me permitió la familiarización con distintas herramientas de Android Studio, así como con los lenguajes y funcionamiento, mediante la programación de diversas aplicaciones.

Una vez finalizado el tutorial, me encontraba en disposición de iniciar el proyecto de la aplicación que sería el centro de mi Trabajo Final de Grado. Pero antes de ello realicé un diseño previo que agilizase la posterior programación.

8.1.1. Diseño previo a la programación

En la elaboración de este diseño decidí buscar aplicaciones similares a la que se iba a desarrollar.

Estas aportaron diferentes ideas como he comentado anteriormente, aunque la complejidad de dichas aplicaciones era mucho mayor que la de Disfruta Cullera, ya que eran de carácter profesional.

Una vez realizado el análisis de aplicaciones similares ya existentes, pasé a elaborar un diseño previo. En este se plantearon aspectos como cuántos menús tendría la aplicación, de qué manera estaría estructurada la información, cómo accedería a la información el usuario... Aspectos que serán explicados en detalle más adelante.

Por otro lado, también tenía claro que quería que mi aplicación se inundara de fotos de la localidad y todas las posibilidades que ofrece. Pues cuando un usuario viera algo que le gustase se interesaría más por visitarlo. Las imágenes utilizadas las conseguí a través de redes sociales como Facebook e Instagram, de la página de la Oficina de Turismo de Cullera o de fotografías de elaboración propia. Además elabore mediante la herramienta Photoshop fondos de diferentes colores para las *activities* informativas, así como un icono atractivo para la aplicación.

Por último, pasé a buscar toda la información necesaria para empezar con la programación, incluyendo datos e imágenes, de forma que me permitiría realizar una programación más rápida y sencilla.

8.2. Estructura de las actividades

A continuación, se abordará la explicación de la estructura de las *activities* elaboradas, explicando tanto su parte gráfica como su parte lógica con el fin de entender de qué manera han sido programadas.

8.2.1. Archivo Strings

Es importante explicar en qué consiste este archivo puesto que ha sido utilizado para el desarrollo de la aplicación.

Este archivo de tipo XML, sirve para almacenar todas las cadenas de caracteres necesarias en un programa. De esta manera, se tienen todos los mensajes de la aplicación almacenados en un archivo. Esto permite referenciar los textos utilizados en los *layouts* a este archivo facilitando los cambios que sean necesarios.

En el proyecto en cuestión, todos los textos utilizados en la parte gráfica de las *activities*, han sido guardados en el archivo `strings.xml` y referenciados posteriormente para su uso.

```
1 <resources>
2   <string name="app_name">Disfruta Cullera</string>
3   <string name="empezar">V I S I T A R \u0020\u0020\u0020\u0020 U L L E R A </string>
4   <string name="contactos">C O N T A C T O S </string>
5
6   <string name="enero">E N E R O </string>
7   <string name="febrero">F E B R E R O </string>
8   <string name="marzo">M A R Z O </string>
9   <string name="abril">A B R I L </string>
10  <string name="mayo">M A Y O </string>
11  <string name="junio">J U N I O </string>
12  <string name="julio">J U L I O </string>
13  <string name="agosto">A G O S T O </string>
14  <string name="septiembre">S E P T I E M B R E </string>
15  <string name="octubre">O C T U B R E </string>
16  <string name="noviembre">N O V I E M B R E </string>
17  <string name="diciembre">D I C I E M B R E </string>
18
19  <string name="ayuntamiento"><b>Ayuntamiento de Cullera</b></string>
20  <string name="tlf_ayun"><b>Teléfono:</b> 961 72 00 00</string>
21  <string name="face_ayun"><b>Facebook:</b> Ajuntament de Cullera</string>
22  <string name="insta_ayun"><b>Instagram:</b> mia_cullera</string>
23  <string name="ofi_turismo"><b>Oficina de turismo</b></string>
24  <string name="tlf_ofiturismo"><b>Teléfono:</b> 961 73 15 86 / Whatsapp: 661 21 31 55</string>
25  <string name="insta_ofiturismo"><b>Instagram:</b> visit_cullera</string>
26  <string name="policia"><b>Policia local Cullera</b></string>
27  <string name="tlf_policia"><b>Teléfono:</b> 961 73 00 01</string>
28 </resources>
```

Ilustración 4. Ejemplo archivo Strings.

Además, este archivo cobra una especial importancia debido a que es el que posibilita que la aplicación desarrollada sea multilingüaje. Puesto que se trata de una aplicación turística es interesante la posibilidad de distribuirla en varios idiomas. Por ello, desarrollé una parte de la aplicación en inglés para exponer qué proceso se debía seguir y demostrar que sería posible la distribución de esta aplicación en distintos idiomas.

Para empezar creé un archivo en la carpeta `values` con el nombre `strings.xml`, el mismo nombre que tenía el archivo en castellano. Tras seleccionar el calificador “local” escogí el idioma inglés para cualquier región. De esta manera, cuando la aplicación se instalara en un dispositivo con el inglés como idioma seleccionado, la aplicación utilizaría este nuevo archivo `strings.xml`.

Una vez creado el nuevo archivo introduce todas las cadenas de texto utilizadas en la aplicación, con los mismos nombres que tenían en el archivo en español, y traduje algunas de ellas para poder realizar la demostración.

8.2.2. Carpeta Drawable

Esta carpeta se encuentra en el directorio `AppTurismoCullera/app/src/main/res`, y es utilizada para almacenar las imágenes que se utilizan en el desarrollo de la aplicación. De esta manera solo hay que referenciarlas usando esta carpeta para que aparezcan en la parte gráfica.

8.2.3. Cambio de letra y sombreado

Antes de empezar a detallar la estructura de las *activities*, explicaré un procedimiento que se ha seguido en la mayoría de estas con el fin de no repetirlo para cada una: El cambio de letra y su sombreado.

Con el fin de conseguir que la aplicación tuviera una apariencia más atractiva, opté por realizar un cambio de letra en los menús. Para ello seguí una serie de procedimientos.

Descargué el archivo `.ttf` de la letra escogida. Dentro del proyecto en cuestión, creé una carpeta denominada *assets* en la ruta: `AppTurismoCullera/app/src/main`. Dentro de esta carpeta creé otra, denominada *fuentes*, que contendría archivo `.ttf` correspondiente a la letra.

En la parte lógica de cada *activity*, es decir en el archivo `.java`, se realizó el cambio de letra. Para ello se referenciaron los botones que había en la parte gráfica correspondiente a cada *activity* dentro del método `onCreate` mediante el método `findViewById`.

Tras ello, se introdujo el tipo de letra. Creé una variable privada de tipo `TypeFace` y pasé a cargar la letra escogida, dentro del método `onCreate`.

```
private Typeface letral;  
  
String fuente1 = "fuentes/letral.ttf";  
this.letral = Typeface.createFromAsset(getAssets(), fuente1);
```

Finalmente, establecí este tipo de letra para cada uno de los botones mediante el método `setTypeFace`, también dentro del método `onCreate`.

Una vez hecho esto en cada una de las *activities* que actuaban como menús conseguí un cambio de letra en este tipo de pantallas que hacían más atractiva la aplicación.

Las *activities* de menús a las que cambié el tipo de letra estaban compuestas por botones. Cada uno de estos botones tenía como fondo una imagen representativa de la *activity* a la que llamaba. Esto suponía un problema ya que, como cada imagen tenía unos colores, el texto no se visualizaba correctamente. De esta manera, decidí añadir sombreado a los textos con la finalidad de que tuvieran una lectura fácil fuera cual fuera el fondo del botón.

Para ello, en la parte gráfica de cada una de las *activities*, hice uso de dos atributos de los botones denominados `shadowColor` y `shadowRadius`.

- `shadowColor`: Define el color que tiene el sombreado, en este caso, al ser el texto blanco, decidí escoger el negro como el color de sombreado.

- `shadowRadius`: Define el tamaño que tendría el sombreado, en este caso, el tamaño que conseguía el efecto buscado, fue de 15.



Ilustración 5. Ejemplo del cambio de letra con sombreado.

8.2.4. Activities

8.2.4.1. *MainActivity*

Esta es la *activity* principal de la aplicación, representa la primera pantalla con la que interactúa el usuario al iniciar la aplicación. En ella el usuario puede acceder a los contactos más importantes de los servicios de la localidad o empezar a buscar información accediendo al menú mensual. Por ello, su finalidad es ser atractiva y lo más sencilla posible.

Parte lógica:

La parte lógica de esta *activity* contiene el método `onCreate` y dos métodos vinculados a los botones existentes en la parte gráfica.

El método `onCreate` se utiliza para realizar el cambio de letra y para vincular la parte lógica con su correspondiente *layout*.

Los métodos vinculados a los botones, consisten en la llamada a otras *activities*, para ello se utiliza un objeto de la clase `Intent` que indica a qué clase llamar, y el método `startActivity` que permite su apertura.

Las clases que se han importado para el funcionamiento de esta *activity* son: `Intent`, `TypeFace`, `AppCompatActivity`, `Bundle`, `View` y `Button`.

Parte gráfica:

Compuesta por un `ConstraintLayout` y dos botones.

- `ConstraintLayout`: Cambio de fondo mediante el atributo `background` por una imagen, contiene los dos botones.
- Botones: Fijados a los extremos de la pantalla para establecer sus posiciones, algunos de los atributos utilizados son:
 - o `onClick`: Permite la vinculación a los métodos creados en la parte lógica.
 - o `background`: Se establece un color semitransparente
 - o `layout_width` y `layout_height`: permiten establecer un tamaño determinado.

Enlaces:

Desde esta pantalla es posible acceder a la pantalla de contactos y al menú de meses.

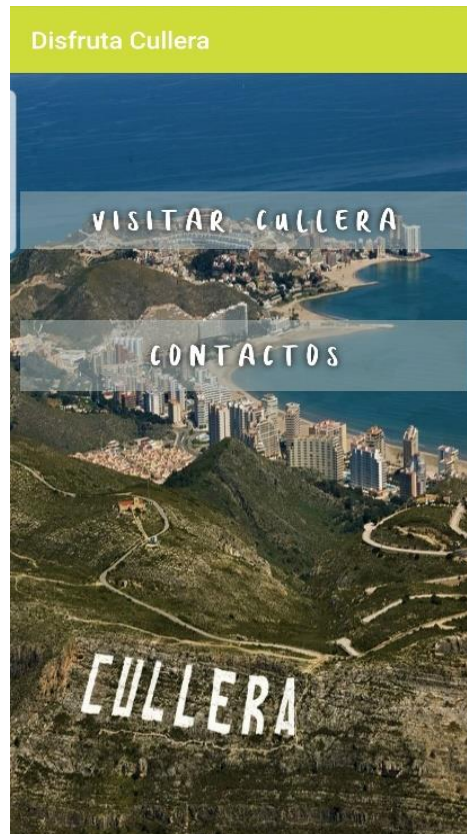


Ilustración 6. Pantalla de inicio de la aplicación.

8.2.4.2. Contactos

Esta *activity* contiene los contactos más útiles para los turistas de los servicios de la localidad: policía, centro de salud, ayuntamiento, guardia civil y oficina de turismo. Además permite acceder a las redes sociales que corresponden a estas instituciones.

Parte lógica:

La parte lógica de esta *activity* contiene la declaración de las cadenas de texto que contienen las URL necesarias para acceder a los sitios web de las instituciones de las que se aporta información. Por otro lado, contiene el método `onCreate` y diversos métodos que permiten llamar a la clase que contiene el `WebView` y enviarle la URL que debe abrir.

El método `onCreate` se utiliza para realizar el cambio de letra y para vincular la parte lógica con su correspondiente *layout*.

Los otros métodos van vinculados a textos que nombran los Facebooks e Instagrams de las instituciones de las que se da contacto. Estos consisten en la llamada a una *activity* que contiene un `WebView`, para ello se utiliza un objeto de la clase `Intent` que indica a qué clase llamar, el método `putExtra` que permite enviar la URL correspondiente a la clase que se ha llamado y el método `startActivity`, que permite su apertura.

Las clases que se han importado para el funcionamiento de esta *activity* son: `Intent`, `AppCompatActivity`, `Bundle` y `View`.

Parte gràfica:

Compuesta por un `ConstraintLayout`, un `LinearLayout` y diversos `TextViews`.

- `ConstraintLayout`: Cambio de fondo por una imagen de elaboración propia que se utilizará como en la aplicación utilizando diversos colores, contiene el `LinearLayout`.
- `LinearLayout`: este tiene orientación vertical y permite establecer los `TextView` necesarios de manera ordenada y simple. Ocupa toda la pantalla y está fijado a los extremos de esta para establecer su posición
- `TextViews`: se utilizan diversos `TextViews` para presentar la información. Algunos atributos utilizados son:
 - o `Gravity`: Para establecer el texto en la parte superior central del espacio que le corresponde se seleccionan las opciones `bottom` y `center_horizontal`.
 - o `TextStyle`: se selecciona la opción `negrita` para los nombres de las instituciones.
 - o `TextIsSelectable`: Este atributo se selecciona en los `TextView` que contienen datos de contacto para que al presionarlos la aplicación permita al usuario copiar la información, acceder al sitio web, llamar al teléfono de forma directa o escribir al correo de forma directa.
 - o `OnClick`: Para los textos que permiten el acceso a la `activity` `SitiosWeb`.

Enlaces:

Desde esta pantalla es posible acceder a la `activity` `SitiosWeb` abriendo los Facebooks e Instagrams disponibles en la información presentada.



Ilustración 7. Pantalla contactos.

8.2.4.3. Menús

La aplicación desarrollada contiene una treintena de menús diferentes que organizan el acceso a la información. La metodología de programación seguida para estos es la misma, por tanto, voy a realizar una explicación general que engloba la estructura que siguen todas las *activities* que actúan como menú.

Parte lógica:

La parte lógica de estas *activities* contiene el método `onCreate` y diversos métodos que permiten llamar a otras *activities*.

El método `onCreate` se utiliza para realizar el cambio de letra y para vincular la parte lógica con su correspondiente *layout*.

Los otros métodos van vinculados a botones que permiten acceder a otros menús o a pantallas informativas. Estos consisten en la llamada a la *activity* correspondiente, para ello se utiliza un objeto de la clase `Intent` que indica a qué clase llamar, el método `putExtra` que permite enviar una variable a la clase que se ha llamado y el método `startActivity`, que permite su apertura.

El envío de la variable se inicia en el primer menú de la aplicación: el menú de meses. En este menú existe un método para cada mes, por lo que se crea una variable de tipo `int` que dependiendo del mes toma un valor del 1 al 12, de forma que permite indicar el mes seleccionado.

Esta variable se utiliza en menús inferiores para seleccionar la información que debe aparecer según el mes que se esté consultando. De esta manera, en estos menús, se utilizan estructuras *if-else* que según el mes llaman a una *activity* u otra, cuando la información cambia según el mes de consulta.

Las clases que se han importado para el funcionamiento de esta *activities* son: `Intent`, `TypeFace`, `AppCompatActivity`, `Bundle`, `Button` y `View`.

Parte gráfica:

Compuesta por un `ConstraintLayout`, un `ScrollView`, en los casos que es necesario, un `LinearLayout` y diversos botones.

- `ConstraintLayout`: Contiene el `ScrollView`, cuando lo hay, o en su defecto el `LinearLayout`.
- `ScrollView`: Este se utiliza en aquellos menús que no caben en la pantalla por lo que se necesita habilitar la función de *scroll*, subir y bajar. De esta manera, el usuario puede desplazar la información con el dedo. Contiene el `LinearLayout`, y está fijado a los extremos de esta para establecer su posición.
- `LinearLayout`: este tiene orientación vertical y permite establecer los botones necesarios de manera ordenada y simple. Ocupa toda la pantalla y, cuando no hay `ScrollView`, está fijado a los extremos de esta para establecer su posición.
- Botones: se utilizan diversos botones. Algunos atributos utilizados son:
 - o `OnClick`: Permite la vinculación a uno de los métodos creados en la parte lógica.
 - o `Background`: Se establecen imágenes, representativas de la información a la que se accede, como fondo de los botones.
 - o `Layout_width`: Se utiliza la opción `match_parent` para que ocupen toda la pantalla.

- Layout_height: Se establece una medida común a todos los botones que componen el menú.

Enlaces:

Desde cada uno de los menús es posible acceder a un menú más concreto o directamente a pantallas informativas.



Ilustración 8. Ejemplo de menú.

8.2.4.4. *Activities informativas*

Muchas de las *activities* que forman la aplicación son informativas, es decir, su fin último es aportar información al usuario sobre un determinado tema. Además la mayoría contienen enlaces que redirigen al usuario, usando un `WebView`, hacia páginas web, enlaces a aplicaciones o redes sociales que completan dicha información. Ya que la estructura de estas se repite, haré una explicación conjunta para todas ellas.

Parte lógica:

La parte lógica de estas *activities* contiene el método `onCreate`. En aquellas que permiten el acceso a páginas web, redes sociales o aplicaciones externas, se incluye a su vez la declaración de los objetos de la clase `String` que contienen las URL necesarias, así como, diversos métodos que permiten llamar a la clase que contiene el `WebView` y enviarle las URL necesarias.

El método `onCreate` se utiliza para vincular la parte lógica con su correspondiente *layout*.

Los otros métodos que permiten la llamada de la clase `SitiosWeb`, van vinculados a textos o botones que nombran las páginas web, redes sociales o aplicaciones a las que se da acceso. Estos consisten en la llamada a una *activity* que contiene un `WebView`, para ello se utiliza un objeto de la clase `Intent` que indica a qué clase llamar, el método `putExtra` que permite enviar la URL correspondiente a la clase que se ha llamado y el método `startActivity`, que permite su apertura.

Las clases utilizadas para el funcionamiento de estas *activities* son: `Intent`, `AppCompatActivity`, `Bundle` y `View`.

Parte gráfica:

La parte gráfica de estas *activities* se basan en el uso de `TextViews` e `ImageViews`. Algunas también contienen botones.

- `ConstraintLayout`: Cambio de fondo por una imagen de elaboración propia en diferentes colores según la temática de la información, contiene el `LinearLayout`.
- `ScrollView`: Este se utiliza en aquellas pantallas donde la información que se da no caben en la pantalla por lo que se necesita habilitar la función de *scroll*, subir y bajar. De esta manera, el usuario puede desplazar la información con el dedo. Contiene el `LinearLayout`, y está fijado a los extremos de esta para establecer su posición.
- `LinearLayout`: este tiene orientación vertical y permite establecer los objetos utilizados de manera ordenada y simple.
- `TextViews`: se utilizan diversos `TextViews` para presentar la información. Algunos atributos utilizados son:
 - o `Layout_gravity`: Para establecer el contenedor de texto en la parte central de la pantalla se selecciona la opción `center`.
 - o `TextStyle`: se selecciona la opción `negrita` para los que contienen los títulos.
 - o `TextIsSelectable`: Este atributo se selecciona para que al presionar datos la aplicación permita al usuario copiar la información, acceder al sitio web, llamar al teléfono de forma directa o escribir al correo de forma directa.

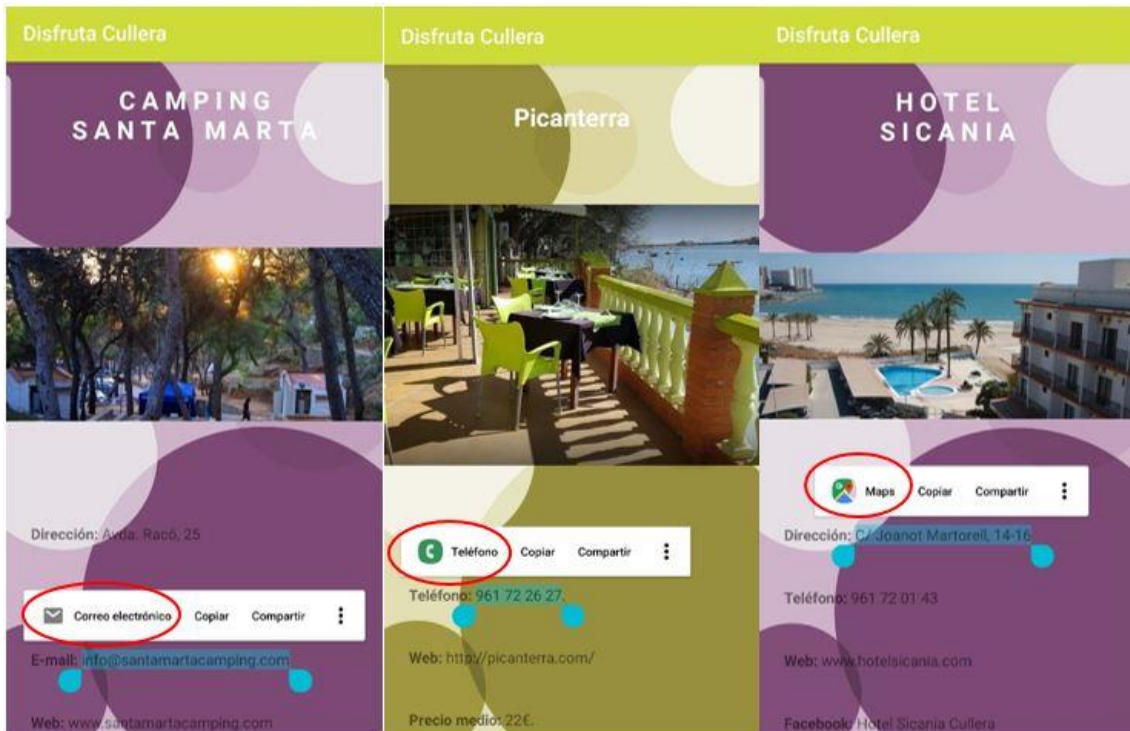


Ilustración 9. Ejemplos del texto seleccionable.

- **OnClick:** Para los textos que tienen vinculado algún método.
- **Layout_width:** Se establece una anchura determinada para los textos dejando márgenes a ambos lados de estos.
- **Layout_height:** En los títulos se establece una altura determinada para el espacio dónde está el título para separarlo del resto de información. Para el resto de textos se establece que se adapte al contenido mediante la opción `wrap_content`.
- **DrawableLeft:** Se utiliza en algunos de los `TextViews` para incluir una imagen a la izquierda del texto.
- **ImageViews:** Es este tipo de objetos se modifican los siguientes atributos:
 - **Layout_width:** Se selecciona la opción `match_parent` con el fin de que ocupe todo el espacio horizontal de la pantalla.
 - **Layout_height:** Se selecciona un valor para cada imagen con el fin de que se visualice correctamente.
- **Botones:** Para los botones se utilizaron los siguientes atributos:
 - **OnClick:** Permite la vinculación a uno de los métodos creados en la parte lógica.
 - **Background:** Se establecen imágenes o colores como fondo de los botones.
 - **Layout_width:** Se utiliza la opción `match_parent` para que ocupen toda la pantalla.
 - **Layout_height:** Se establece una medida que posibilite un diseño atractivo.
 - **DrawableLeft:** Se utiliza en algunos de los `TextViews` para incluir una imagen a la izquierda del texto.

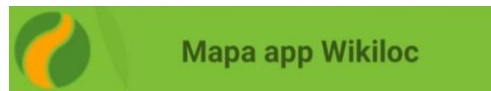


Ilustración 10. Ejemplo de uso del atributo leftDrawable.

Enlaces:

Desde las pantallas de carácter informativo se puede acceder a los sitios web, aplicaciones o redes sociales que son nombrados. Además algunas de estas pantallas tienen acceso a mapas que dan más información.

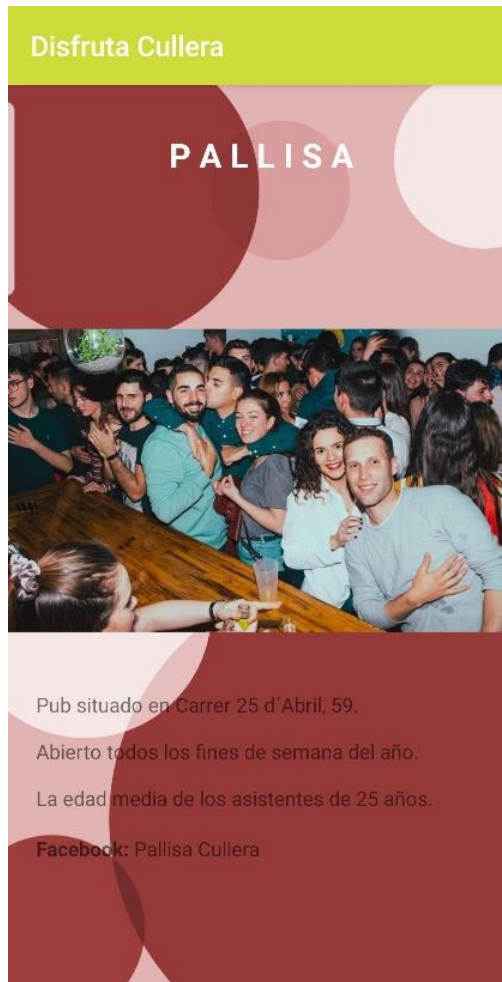


Ilustración 11. Ejemplo de activity informativa.

8.2.4.5. Mapas

Estas *activities* se utilizan para dar indicaciones sobre un lugar, una zona o un recorrido. Para ello se crea una *Google Maps Activity*.

Para el uso de los mapas de google es necesaria la instalación de Google Play Services. Por otra parte también es necesario incluir en el fichero `AndroidManifest.xml` el permiso `android.permission.ACCESS_FINE_LOCATION`.

En los diferentes mapas que he utilizado en el desarrollo de la aplicación se ha hecho uso de marcadores para indicar lugares concretos, líneas para indicar la ruta seguida por el carril bici y polígonos para resaltar zonas en las pantallas de aparcamiento.

Para ello busqué las coordenadas de los lugares y los puntos que iba a utilizar mediante Google Maps y las almacenaba en variables tipo `LatLng`. Para la creación de líneas y polígonos hacía uso de variables de tipo `Polyline` y `Polygon`, mientras que para los *markers* utilizaba variables de tipo `marker`.

Para que al abrir los mapas estos enfocaran directamente en la zona de Cullera utilizaba el método `moveCamera` a las coordenadas del centro del pueblo y un `zoom` de 12.

Para crear los polígonos utilizaba el método `addPolygon` utilizando los puntos que lo delimitaban. Además establecí un color para el borde y un color semitransparente para el relleno con la finalidad de que se vieran las calles bajo los polígonos.

Para la creación de líneas utilicé el método `addPolyline` utilizando los puntos que definían el recorrido y estableciendo un color para ella.

Por último, para agregar los marcadores utilicé el método `addMarker`, para ello se necesita crear una variable `MarkerOptions` en la que se indica la posición, el título, una descripción en algunos y el color de cada marcador.

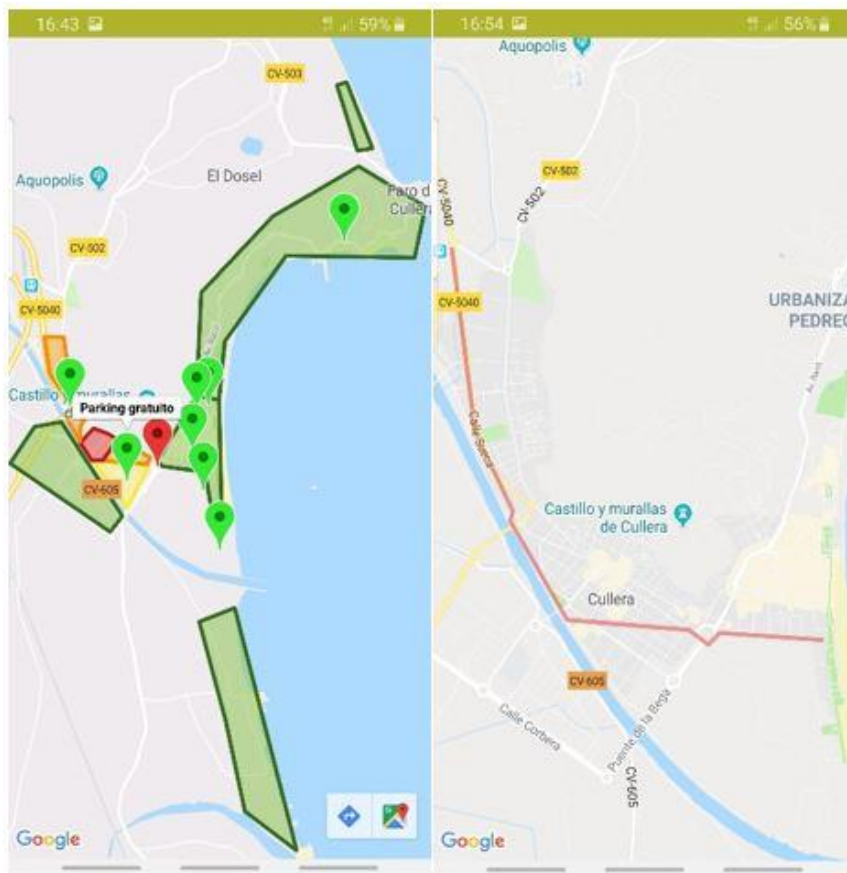


Ilustración 12. Ejemplos de mapas.

8.2.4.6. *SitiosWeb*

Esta *activity* se utiliza para acceder a todos los sitios web, redes sociales y aplicaciones externas a las que da acceso la aplicación.

Parte lógica:

Se crea una variable del tipo `WebView` y se enlaza al `WebView` existente en el *layout* mediante el método `findViewById`.

Se recupera la URL que se le envía mediante el método `getStringExtra` y tras ello se procede a cargar esta:

```
wv1.getSettings().setJavaScriptEnabled(true);  
wv1.setWebViewClient(new WebViewClient());  
wv1.loadurl(url);
```

Para la funcionalidad de esta *activity* es necesario importar las siguientes clases: `AppCompatActivity`, `Bundle`, `WebView` y `WebViewClient`.

Parte gráfica:

Se compone por un `ConstraintLayout` que contiene un `WebView` ocupando toda la pantalla y fijado a los extremos para establecer su posición.

Enlaces:

Esta *activity* no lleva a ninguna otra, únicamente permite la visualización de las URL que se le envían.

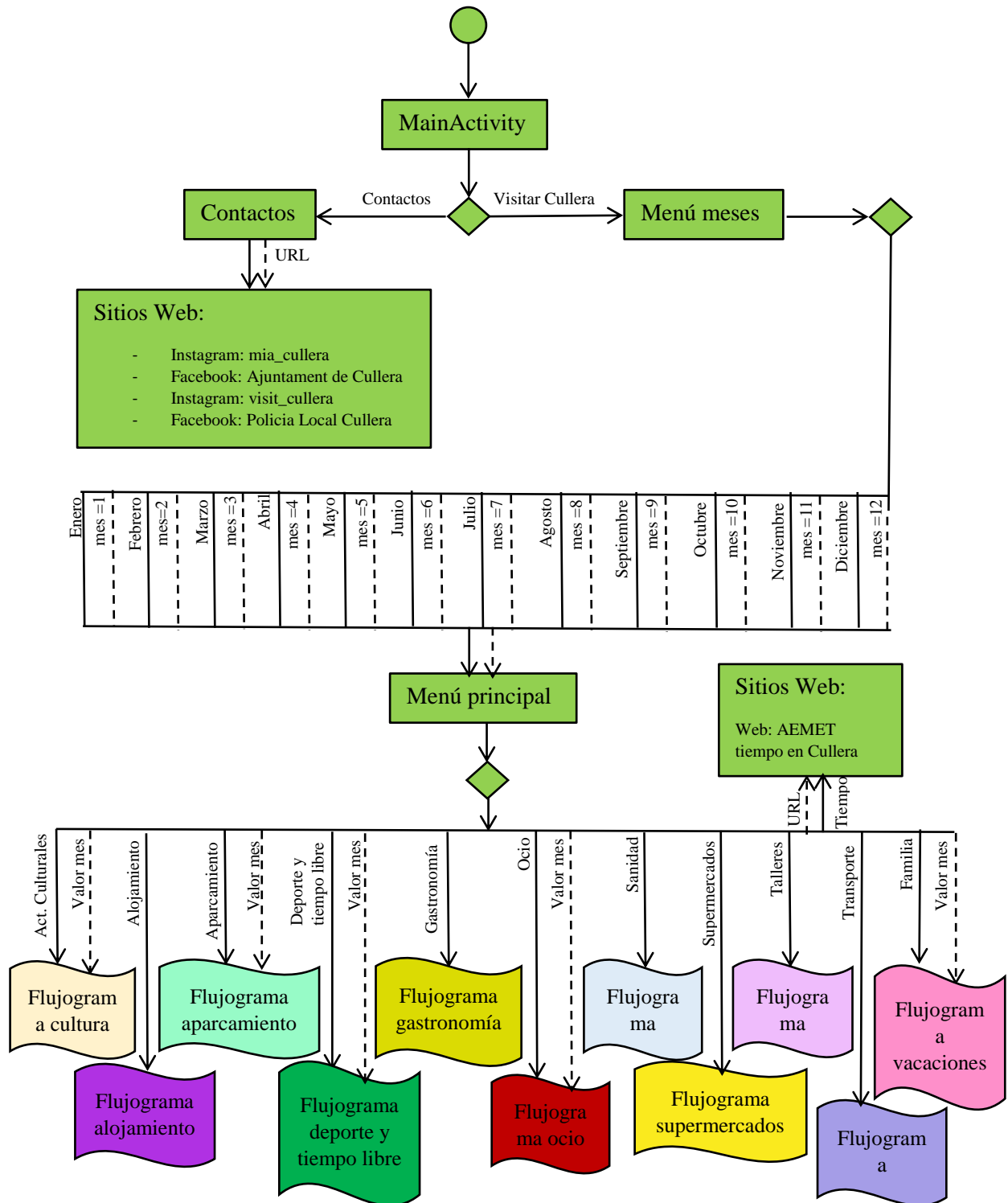
8.3. Funcionamiento de la aplicación: Flujogramas y explicación

Tras la explicación de cómo se ha programado cada una de las *activities* que componen la aplicación se detallará la interrelación que hay entre ellas y el funcionamiento de la aplicación. Para ello se hará uso de flujogramas.

Debido a la gran cantidad de *activities* que componen la aplicación, más de 120, resultaba complicado elaborar un flujograma conjunto para todas ellas, por lo que se realizarán diversos flujogramas relacionados entre sí para facilitar su comprensión, acompañados de una explicación que complementará la información.

Para empezar elaboraré un flujograma que irá desde la pantalla principal de la aplicación hasta el menú principal de la misma.

8.3.1. Flujograma principal



Flujograma 1. Principal.



Este flujograma presenta las primeras pantallas con las que interactúa el usuario al iniciar la aplicación. La primera de ellas es la correspondiente a la *activity* `MainActivity`, en la que se ofrecen dos opciones: consultar los contactos de las principales instituciones o empezar con la guía de turismo.

Al acceder a la *activity* `Contactos`, el usuario, aparte de obtener información puede acceder a los sitios web que se ofrecen clicando en los mismos, para ello se hace uso de la *activity* `SitiosWeb`, a la que se envían las URL pertinentes. Los sitios web a los que se facilita el acceso son: Instagram y Facebook del ayuntamiento, Instagram de la oficina de turismo y Facebook de la Policía Local.

Por otro lado, si se opta por empezar con la guía de turismo, el usuario accede a la *activity* `MenuMeses`, en esta se encuentra un listado de todos los meses, que permite acceder a una información u otra dependiendo del mes seleccionado. En esta *activity* se crea una variable que adoptará un valor u otro según el mes y permitirá la selección de información en posteriores *activities*.

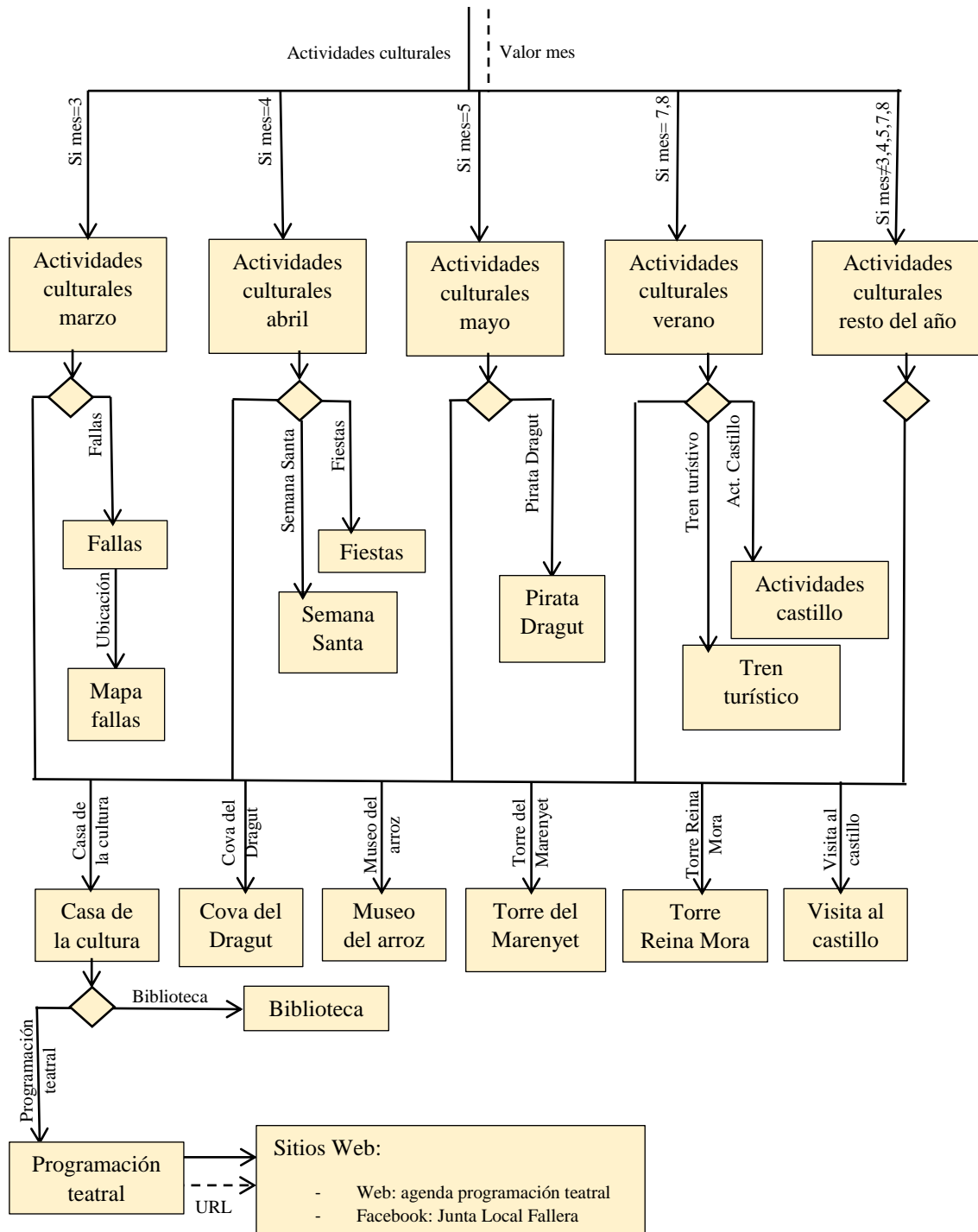
Tras la selección de un mes, el usuario llegará a la *activity* `MenuPrincipal`, a la que le llegará el valor de la variable. Una vez en esta *activity*, el usuario podrá seleccionar entre la distinta información que se proporciona: actividades culturales, alojamiento, aparcamiento, deporte y tiempo libre, gastronomía, ocio, sanidad, supermercados, talleres, tiempo, transporte y vacaciones en familia.

A las *activities* que se ocupan de estas temáticas también se enviará el valor de la variable `mes`, siempre y cuando la información que estas expongan cambie según el periodo del año en el que se visita la localidad, si la información no varía, la variable `mes` no es necesaria.

La opción tiempo dirige al usuario a la *activity* `SitiosWeb`, en la que se consulta el tiempo en la localidad de Cullera, mediante el envío de la URL necesaria.

A partir de este punto, expondré diversos flujogramas para cada una de estas temáticas, con la finalidad de que la información expuesta se vea de una forma más clara.

8.3.2. Flujograma actividades culturales



Flujograma 2. Actividades culturales.

En este flujograma se presentan todas las *activities* que contienen información cultural. Al seleccionar la opción de consultar actividades culturales, la app envía al usuario a 5 pantallas diferentes dependiendo del mes seleccionado previamente, mediante bucles *if - else* condicionados por el valor de la variable *mes*.

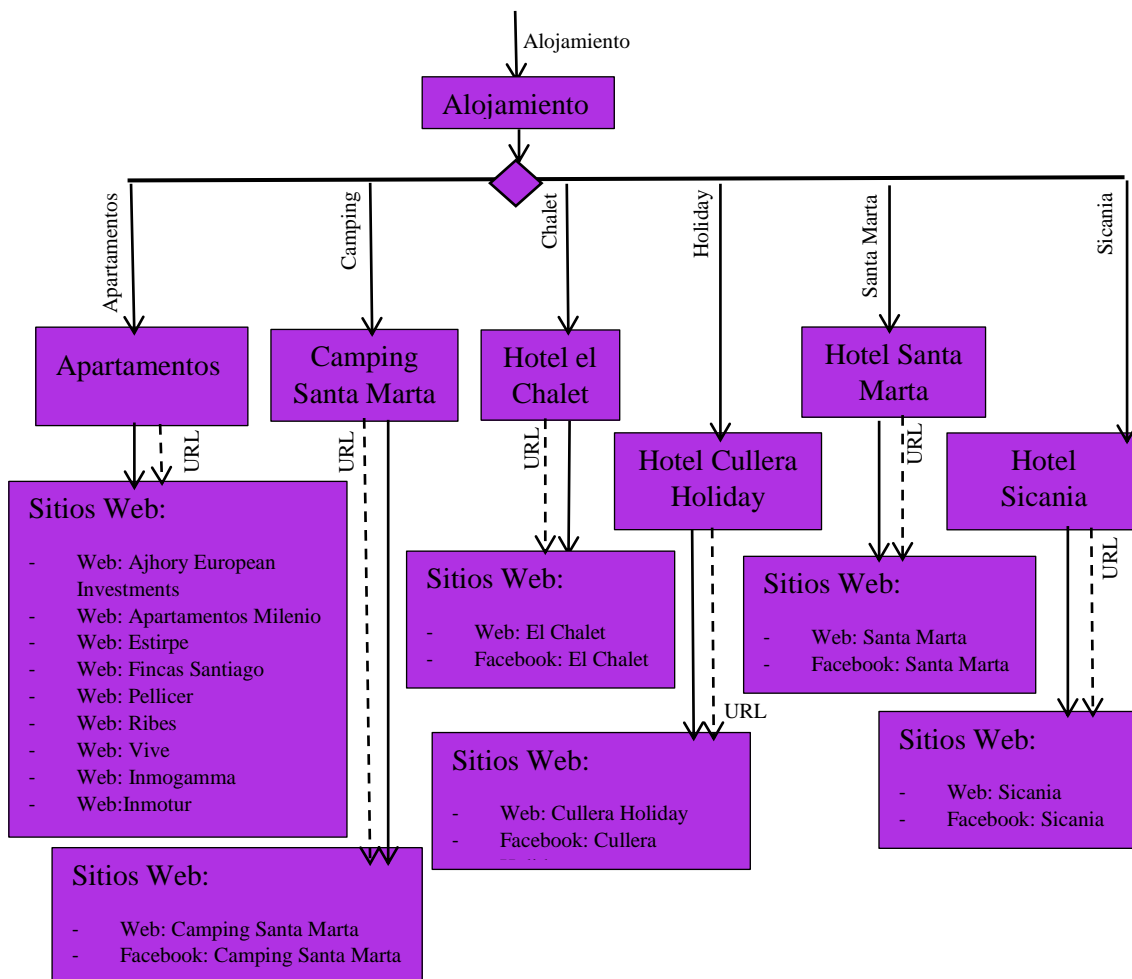
Estas pantallas actúan como menús que comparten algunas actividades que se realizan durante todo el año, y cada una de ellas contiene alguna actividad específica para el mes seleccionado en concreto.

En estos menús se puede acceder a gran variedad de actividades relacionadas con la cultura. Para cada actividad existe una *activity* de carácter informativo que presenta toda la información necesaria para conocerla y realizarla o visitarla. Dos de estas *activities*, para completar la información expuesta, enlazan con otras.

La *activity* Fallas contiene un botón que permite acceder a la *activity* de Google Maps MapaFallas en la que se puede ver mediante *markers* dónde se encuentran cada una de las comisiones falleras de la localidad.

La *activity* CasaDeLaCultura actúa como un pequeño menú que da acceso a las *activities* Biblioteca y ProgramaciónTeatral. La primera de estas informa sobre los servicios que ofrece la biblioteca municipal, y la segunda sobre la programación teatral que se ofrece en el salón de actos. Esta última también permite el acceso a la *activity* SitiosWeb, a la que se envían las URL correspondientes, consultando la agenda teatral anual y el Facebook de Junta Local Fallera donde se puede encontrar información acerca del concurso de teatro fallero.

8.3.3. Flujograma alojamiento



Flujograma 3. Alojamiento.

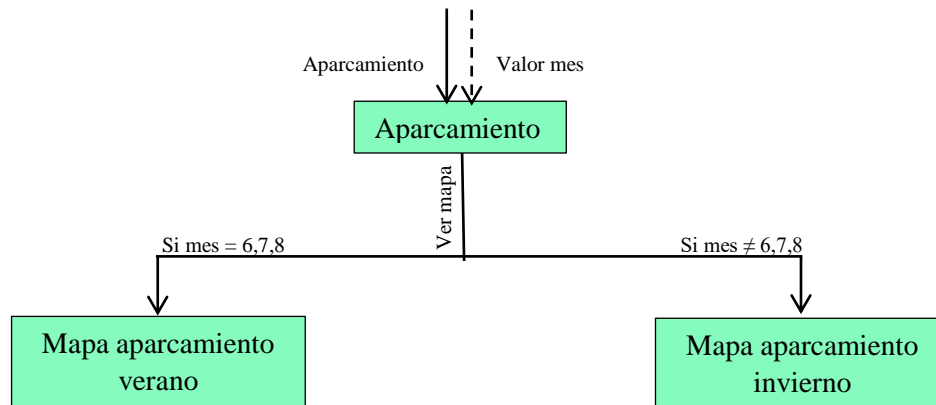
El flujograma anterior expone todas las *activities* que componen la sección de alojamiento de la app. Para esta sección no es necesario recibir el valor de la variable *mes*, pues los alojamientos están operativos durante todo el año.

La *activity* Alojamiento actúa como un menú que contiene los distintos alojamientos que he decidido incluir en la app, que no abarcan todos los disponibles en Cullera, pero sí los más importantes. Además, se incluye una *activity* que contiene datos de contacto de las inmobiliarias más importantes de la localidad.

Respecto a los alojamientos, cuando el usuario escoge cada uno de ellos accede a *activities* de carácter informativo. Cada una de estas permite el acceso a la *activity* SitiosWeb que, mediante un WebView, permite la visualización de las webs y cuentas de Facebook de cada uno de los alojamientos, enviando a esta las URL pertinentes.

La *activity* que contiene los datos de las inmobiliarias también permite el acceso a sus sitios web, enviando las URL de estos a la *activity* SitiosWeb.

8.3.4. Flujograma aparcamiento



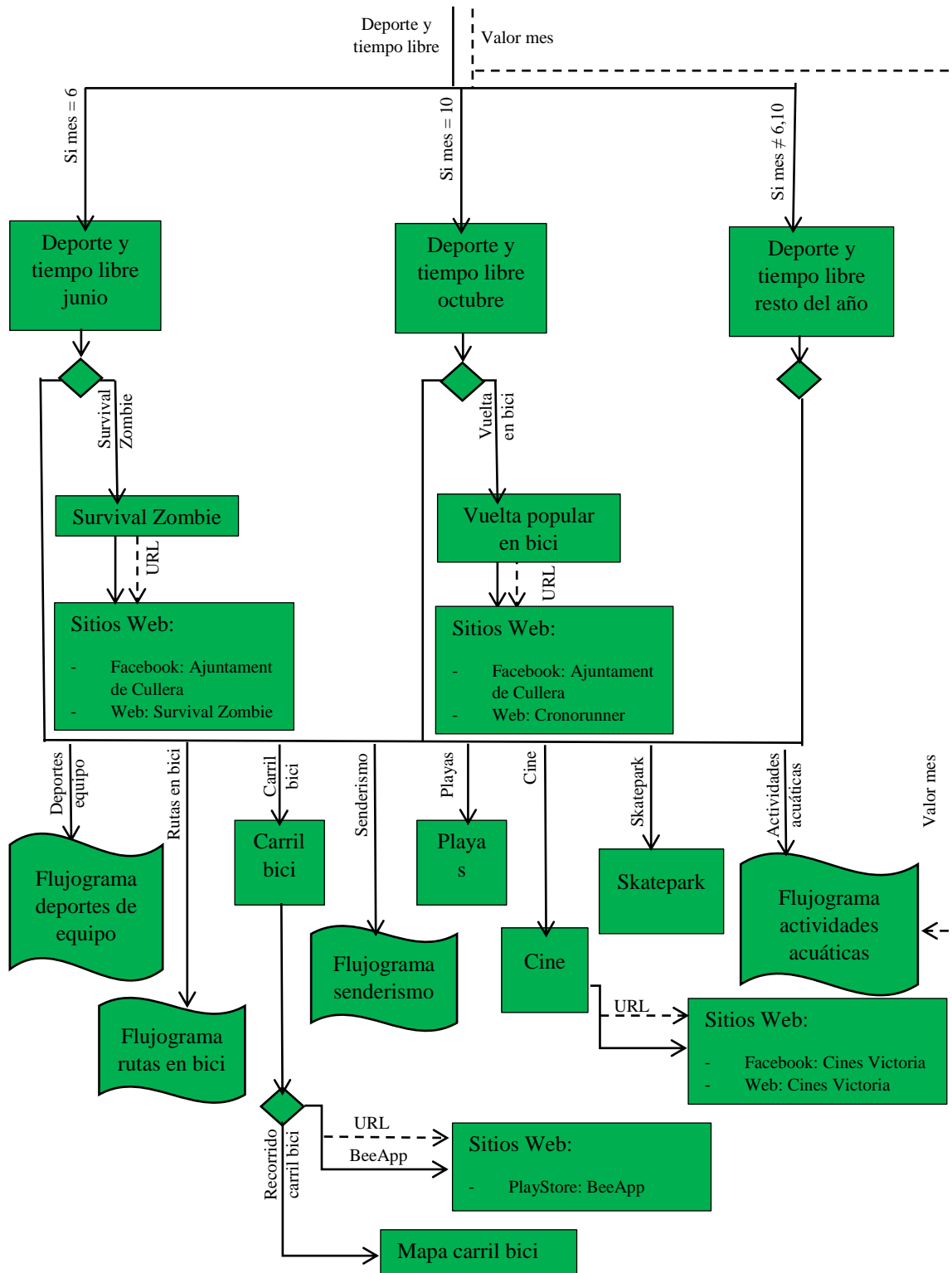
Flujograma 4. Aparcamiento.

En este flujograma explico cómo llega el usuario a la información sobre aparcamiento que se proporciona en la aplicación. A la *activity* Aparcamiento se llega a través de *MenuPrincipal* y se le envía el valor de la variable *mes*.

En esta *activity* se explican los distintos colores que toman las zonas en los mapas, dependiendo del grado de dificultad para encontrar aparcamiento y de si esta es de pago o no. Tras ello, mediante un botón el usuario llega a una *activity* de Google Maps dependiendo del mes que se esté consultando: *MapaAparcamientoVerano* y *MapaAparcamientoInvierno*.

En ambas *activities* se utilizan polígonos para delimitar las zonas para aparcar y la dificultad que estas soportan.

8.3.5. Flujograma deporte y tiempo libre



Flujograma 5. Deporte y tiempo libre.

Este flujograma expone el funcionamiento de la parte de la app que contiene la información sobre deporte y tiempo libre. Puesto que esta parte de la aplicación está compuesta por muchas *activities* utilizaré diversos flujogramas para su explicación.

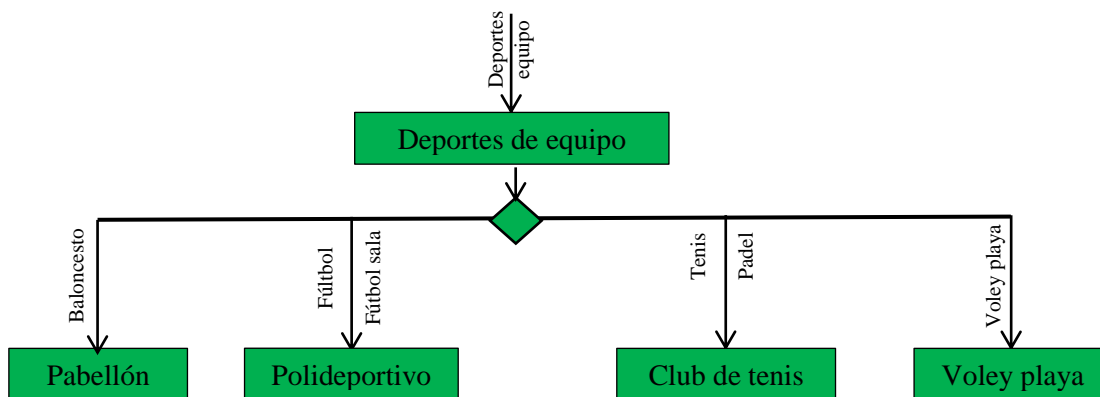
En este primer flujograma, explico el proceso seguido al seleccionar la opción `DeporteYTiempoLibre` desde `MenuPrincipal`. Tras esta selección, el usuario accederá a 3 *activities* diferentes según el mes que esté consultando. Estas actuarán como menús que tendrán actividades en común y actividades únicas, propias del mes en cuestión.

En las *activities* a las que se accede mediante los 3 menús de `DeporteYTiempoLibre`, se puede encontrar información sobre la vuelta popular en bici, la competición *Survival Zombie*, deportes de equipo que se pueden practicar en la localidad, rutas en bici y de senderismo, el carril bici, el cine, las playas, el *Skatepark* y las actividades acuáticas ofrecidas.

Esta información se organiza mediante *activities* que actúan como menú o tienen carácter informativo. Las *activities* de carácter informativo tienen, en su mayoría, acceso a la *activity* `SitiosWeb`, a la que se le envían las URL necesarias. Una de ellas, `CarrilBici`, tiene acceso además a una *activity* de Google Maps, `MapaCarrilBici`, en la que se ve el recorrido de dicho carril que se ha realizado mediante líneas.

Las *activities* que funcionan como menú desagregan la información en diferentes *activities* informativas y se explicarán en los flujogramas posteriores, puesto que no era posible compactar todas las *activities* en un mismo flujograma.

8.3.5.1. Flujograma deportes de equipo

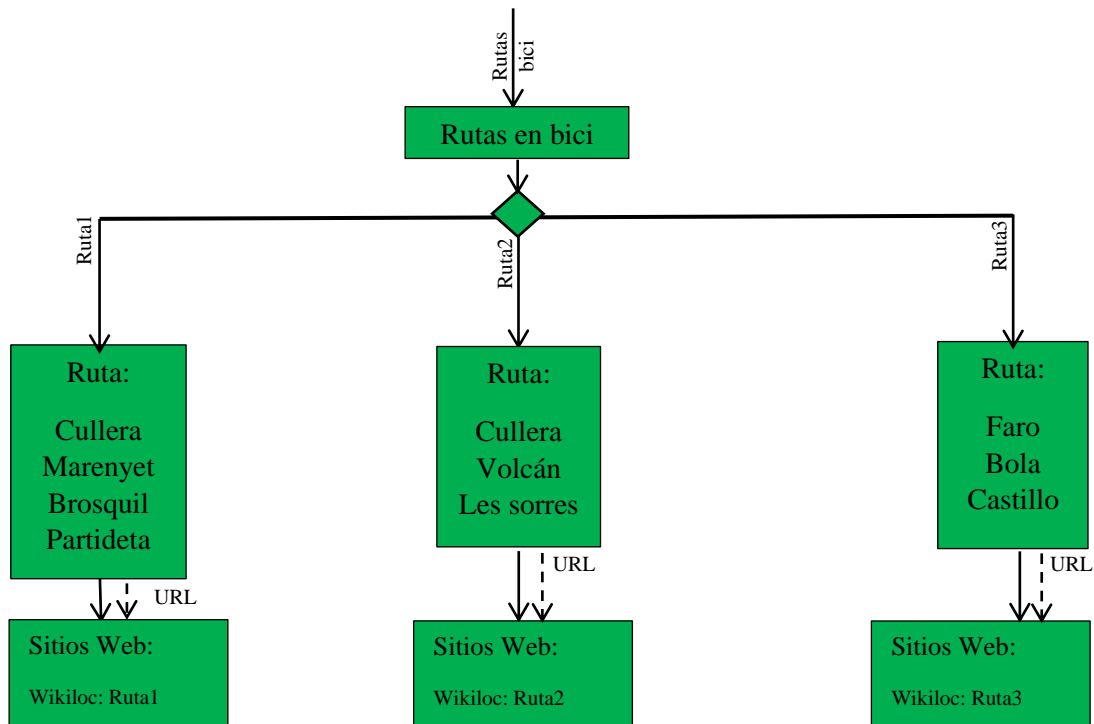


Flujograma 6. Deportes de equipo.

El flujograma anterior expone la relación de *activities* que informan sobre la posibilidad de practicar deportes de equipo en Cullera. Cuando el usuario escoge la opción deportes de equipo pasa a un menú que contiene todos los deportes que se pueden practicar en instalaciones de la localidad.

Mediante este menú se puede consultar información sobre dónde practicar cada deporte e información para ponerse en contacto con la institución que gestiona estas instalaciones. Algunos de los deportes se practican en las mismas instalaciones y, por tanto, al seleccionarlos, la aplicación conduce al usuario a la misma *activity*.

8.3.5.2. Flujograma rutas en bici



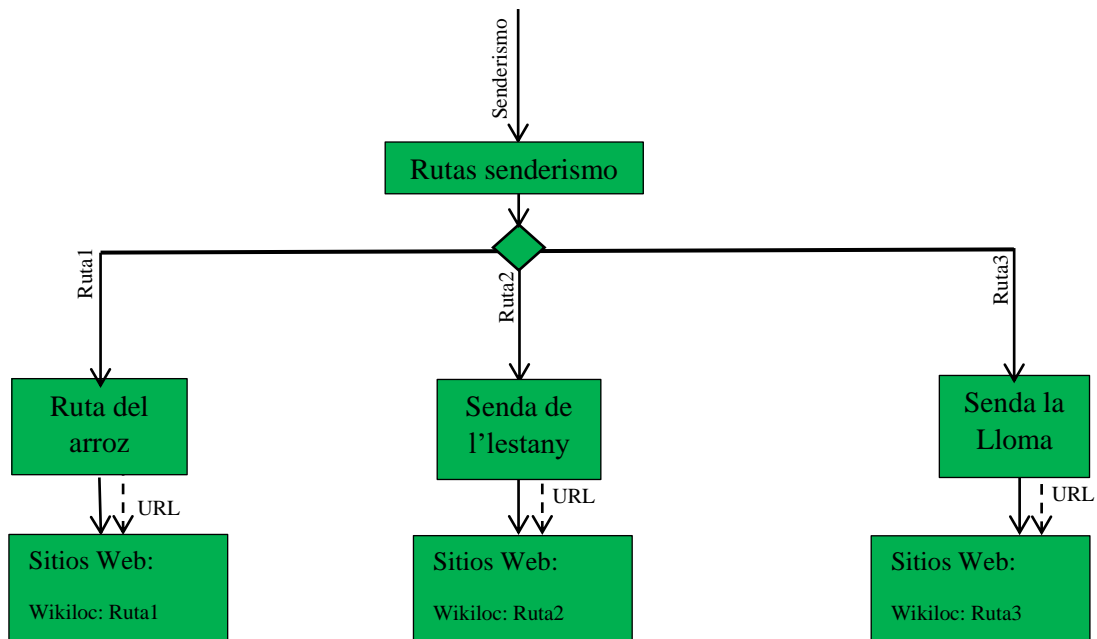
Flujograma 7. Rutas en bici.

Este flujograma presenta las *activities* a las que accede el usuario al solicitar información sobre rutas en bici. En primer lugar, accede a una *activity* que actúa como menú: `RutasEnBici`. Mediante esta podrá acceder a 3 *activities* que presentan información sobre 3 rutas en bici diferentes.

Cada una de estas *activities* presenta acceso a la *activity* `SitiosWeb`, a la que se le envían las URL correspondientes. Mediante estas se permite consultar, en la web o aplicación Wikiloc⁶, información detallada sobre dichas rutas.

⁶ Wikiloc: Mashup gratuita que almacena y comparte rutas al aire libre georreferenciadas y puntos de interés de todo el mundo. Las rutas se organizan en categorías según las características de estas. [26]

8.3.5.3. *Flujograma senderismo*

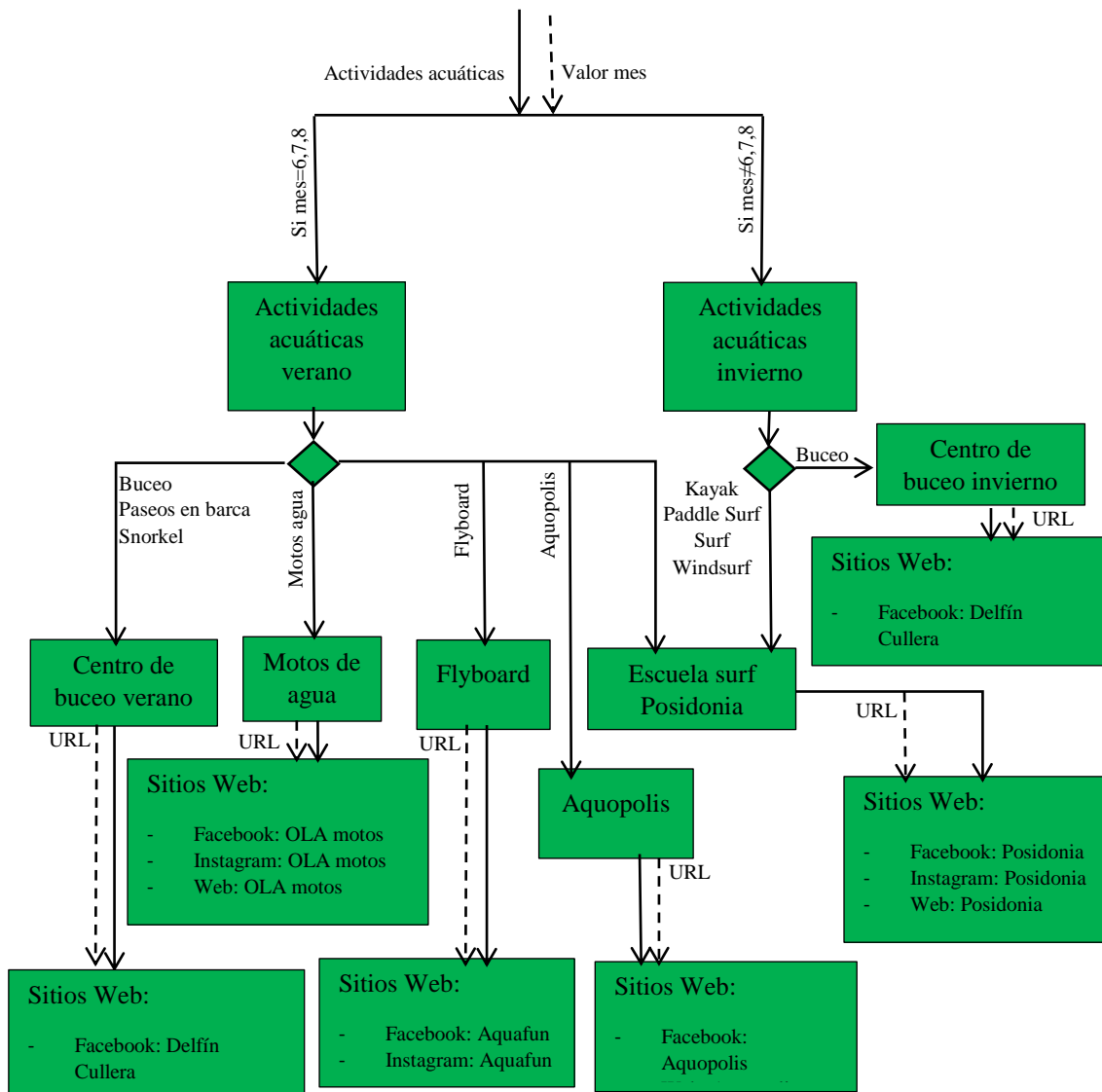


Flujograma 8. Senderismo.

Este flujograma, análogamente al anterior, expone cómo se accede a la información de senderismo. En primer lugar, el usuario accederá a una *activity* que actúa como menú: *RutasSenderismo*. Mediante este podrá acceder a 3 *activities* que presentan información sobre 3 rutas diferentes.

Cada una de estas *activities* presenta también acceso a la *activity* *SitiosWeb*, a la que se le envían las URL correspondientes. Mediante estas se permite consultar, utilizando la web o aplicación Wikiloc de nuevo, información detallada sobre dichas rutas.

8.3.5.4. Flujograma actividades acuáticas



Flujograma 9. Actividades acuáticas.

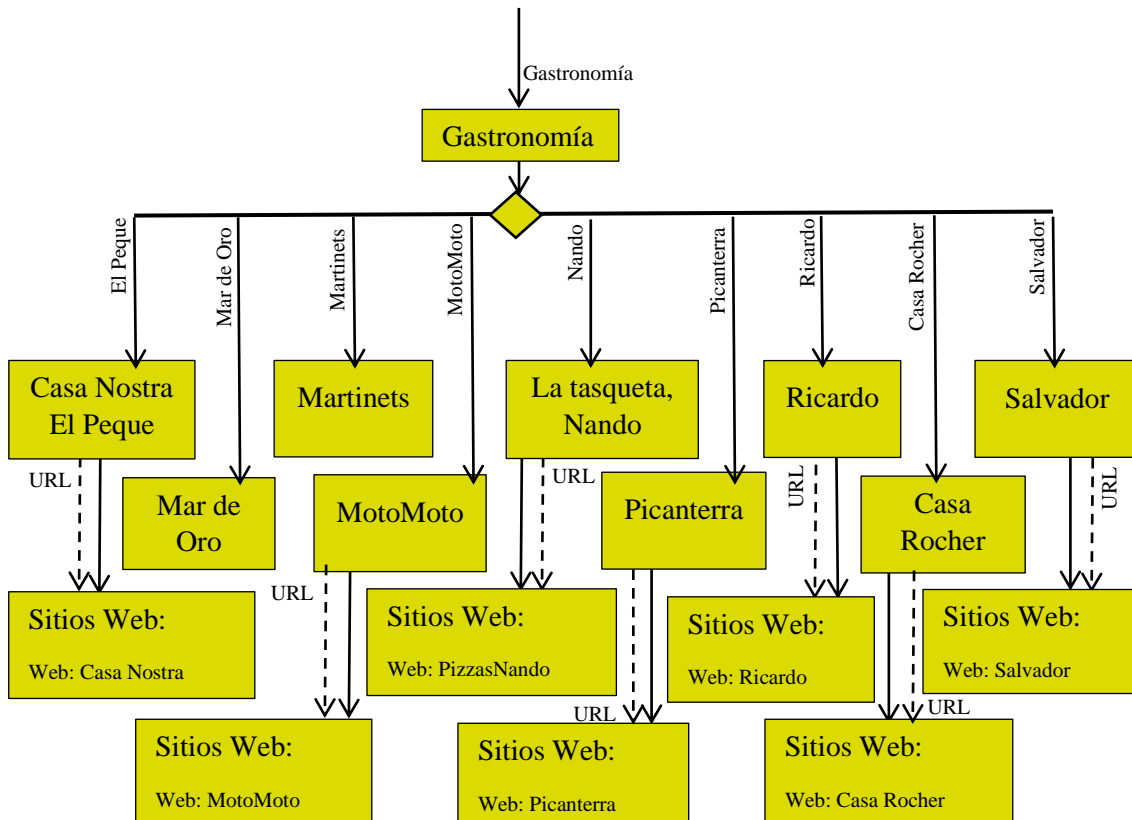
Por último, en lo que respecta a deporte y tiempo libre, se presenta el flujograma de actividades acuáticas. Al seleccionar esta opción en el menú `DeporteYTiempoLibre`, se hace uso de nuevo de la variable `mes`. Dependiendo de esta, se accederá a dos *activities* diferentes que actuarán como menú: `ActividadesAcuáticasVerano` y `ActividadesAcuáticasInvierno`.

Respecto a las de invierno solo se accede a 2 *activities* diferentes, puesto que las actividades kayak, surf, paddle surf y windsurf se realizan en la misma empresa y se utiliza, por tanto, una misma *activity* para ellas. Además, esta *activity* se utiliza también para verano puesto que las actividades que ofrece esta empresa se realizan durante todo el año.

Sin embargo, la opción de buceo lleva a *activities* diferentes según sea verano o invierno, puesto que dicha empresa ofrece más actividades en verano y se utiliza para estas una *activity* que las englobe.

Todas las *activities* que ofrecen información sobre actividades acuáticas permiten acceso a la *activity* SitiosWeb. Clicando en los enlaces que aparecen en las pantallas, se envían las URL necesarias y se accede a los sitios web o redes sociales disponibles.

8.3.6. Flujograma gastronomía

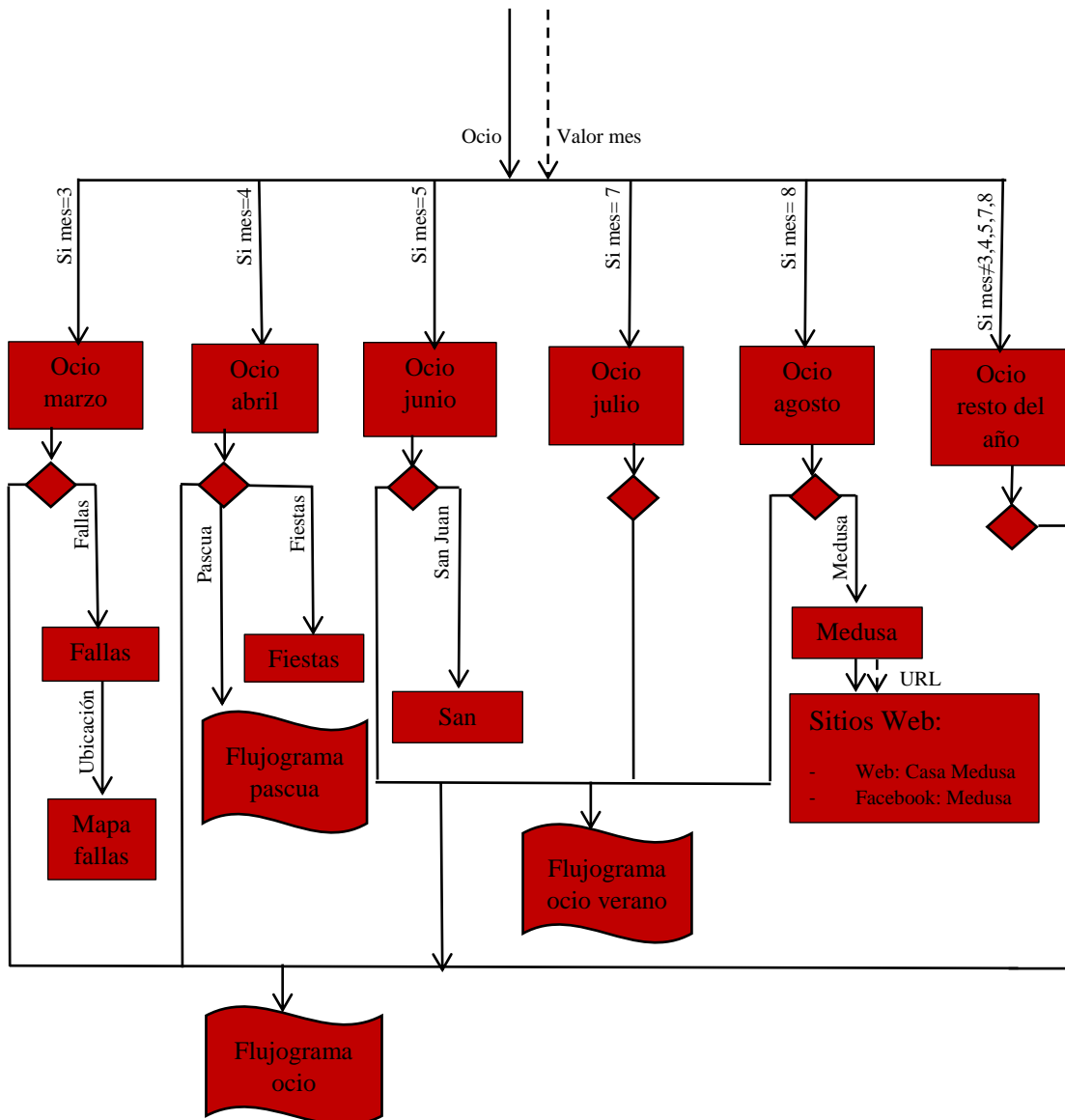


Flujograma 10. Gastronomía.

Este flujograma presenta todas las *activities* que aportan información sobre restaurantes en la app elaborada. Cuando el usuario escoge esta opción en *MenuPrincipal* accede a una *activity* que actúa también como menú, *Gastronomia*. A esta *activity* no se le envía la variable *mes*, puesto que su contenido no varía dependiendo de la época del año.

Mediante el menú *Gastronomia* se accede a varias *activities* de carácter informativo que pertenecen a diversos restaurantes de la localidad. En estas se da información de contacto y, a parte, se permite el acceso a los sitios web de aquellos restaurantes que disponen de él. Para ello, se hace uso de la *activity* *SitiosWeb*, a la que se envía la URL correspondiente en cada caso.

8.3.7. Flujograma ocio



Flujograma 11. Ocio.

Este flujograma presenta todas las *activities* a las que puede acceder el usuario al consultar la información de ocio desde *MenuPrincipal*. Esta parte de la app también la dividiré en diversos flujogramas puesto que está compuesta por muchas *activities*.

Cuando el usuario escoge consultar el ocio de la localidad puede acceder a 6 *activities* diferentes que funcionan como menús de ocio dependiendo del mes consultado, estas se diferencian en una o varias actividades que solo están disponibles en unos meses en concreto.

Las actividades específicas en cada mes son de carácter informativo. Una de ellas, *Fallas*, tiene acceso a una *activity* de tipo Google Maps, *MapaFallas*, en la que se puede ver la localización de las comisiones mediante *markers*.

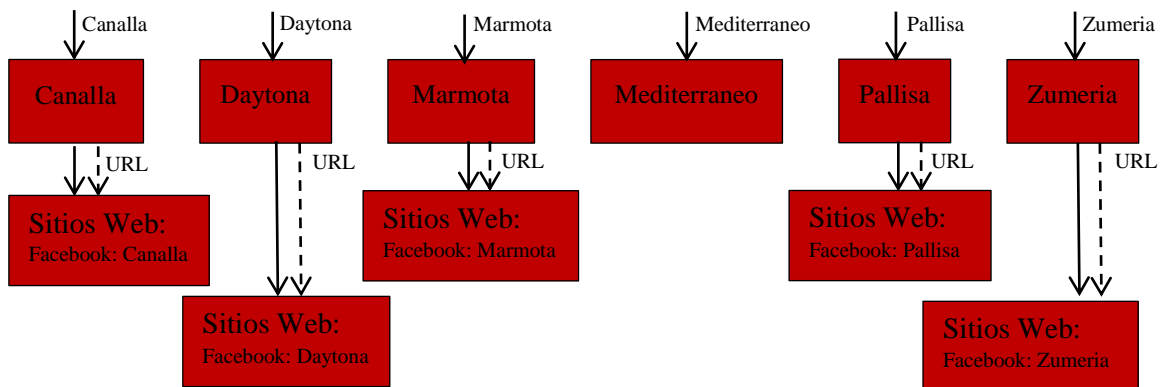
Otra, Pascua, aparte de tener carácter informativo, dispone de un menú que accede a información sobre dónde ir en pascua. Dicho menú se explicará posteriormente mediante un flujograma.

La *activity* específica de agosto, Medusa, da acceso, mediante SitiosWeb, al Facebook y web de la organización del festival.

Por otro lado, los meses de verano, junio, julio y agosto, el menú al que se accede contendrá también todos los locales de ocio de verano, que se verán en un flujograma que presentaré posteriormente.

La parte de ocio de invierno, que se verá en el siguiente flujograma, está presente siempre, independientemente del mes seleccionado, pues son locales que se encuentran abiertos durante todo el año.

8.3.7.1. Flujograma ocio invierno

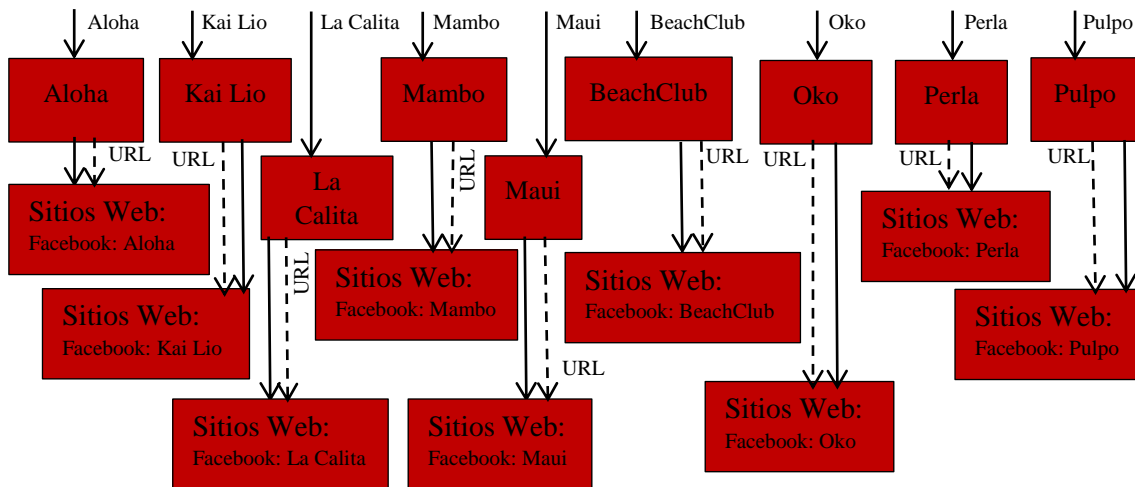


Flujograma 12. Ocio invierno.

Este flujograma presenta las *activities* que contienen información sobre los locales de ocio que están abiertos durante todo el año. El usuario podrá acceder a estas *activities* independientemente del mes que haya seleccionado, puesto que en los 5 menús están disponibles.

Cada una de estas *activities* tiene carácter informativo. Además, mediante estas, el usuario puede acceder a la *activity* SitiosWeb, a la que se envía la URL necesaria y permite visitar el Facebook de cada uno de los locales.

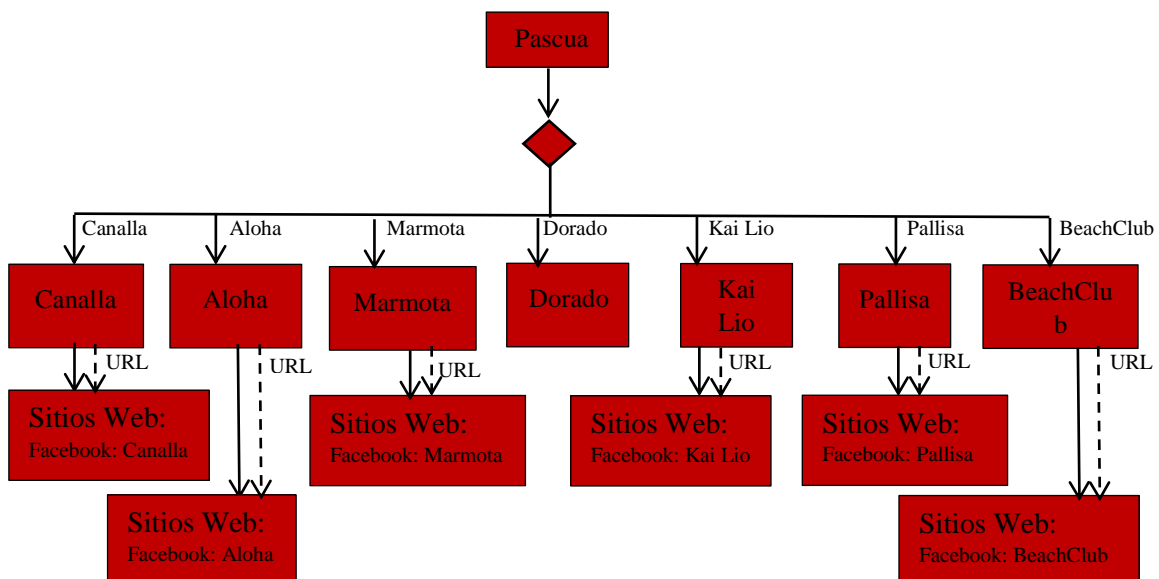
8.3.7.2. *Flujograma ocio verano*



Flujograma 13. Ocio verano.

Mediante este flujograma presento aquellas *activities* que están disponibles en el menú de ocio únicamente los meses de junio, julio y agosto. Cada una de estas *activities* tiene carácter informativo, y el usuario puede visitar el Facebook de los locales mediante la *activity* SitiosWeb, a la que se envía la URL necesaria.

8.3.7.3. *Flujograma Pascua*

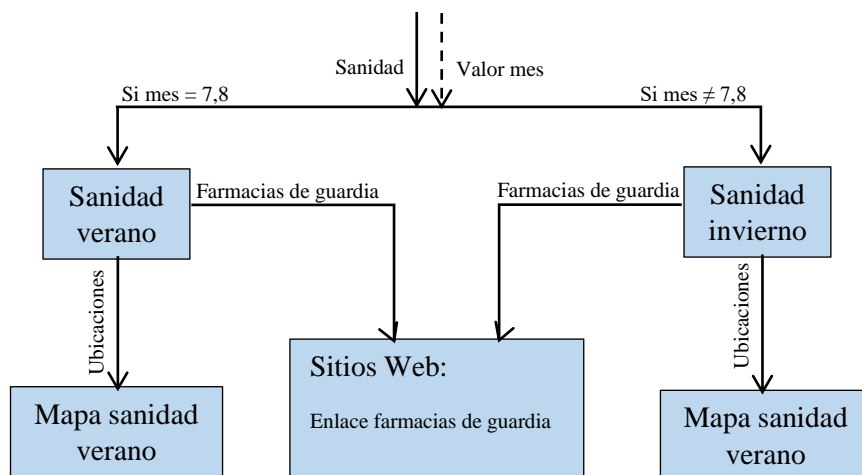


Flujograma 14. Ocio Pascua.

En este último flujograma de la parte de ocio, se puede ver la estructura que sigue la *activity* Pascua. Al seleccionar esta opción en el menú de ocio, el usuario encuentra un texto informativo y más abajo un menú que permite acceder a cada una de las *activities* que informan sobre los locales más frecuentados el día de pascua.

Estas *activities* son las mismas que se han presentado en los flujogramas de ocio anteriores, por tanto, tienen las mismas características: carácter informativo, y posibilidad de visitar el Facebook mediante la *activity* SitiosWeb, a la que se envía la URL necesaria.

8.3.7.4. Flujograma sanidad



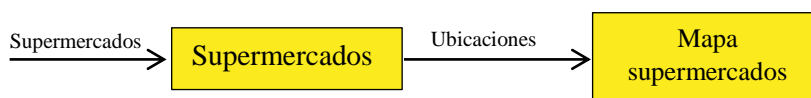
Flujograma 15. Sanidad.

Este flujograma presenta los caminos que sigue la app cuando el usuario consulta la información de sanidad. Existen dos *activities* de sanidad: *SanidadVerano* y *SanidadInvierno*. Dependiendo del mes de consulta la aplicación conducirá al usuario a una u otra.

En estas se proporciona información acerca de sanidad, veterinarios y farmacias. Para completar la misma se permite el acceso, desde cada una de las *activities*, a *activities* de Google Maps, que mediante un mapa, indican las localizaciones de los centros disponibles.

Además, se proporciona un enlace, común a todos los meses del año, que redirecciona al usuario, mediante la *activity* SitiosWeb, a la página dónde se puede encontrar el horario de las farmacias de guardia de la localidad.

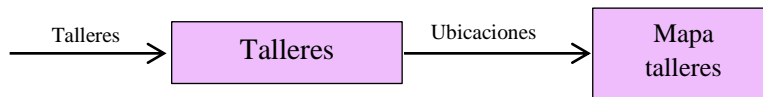
8.3.8. Flujograma supermercados



Flujograma 16. Supermercados.

La *activity* Supermercados es una de las más sencillas de la aplicación. Al seleccionar la opción supermercados en `MenuPrincipal`, el usuario accede a una *activity* informativa, Supermercados, que permite el acceso a una *activity* de Google Maps, `MapaSupermercados`, dónde, mediante *markers*, se indica la posición de los supermercados de la localidad.

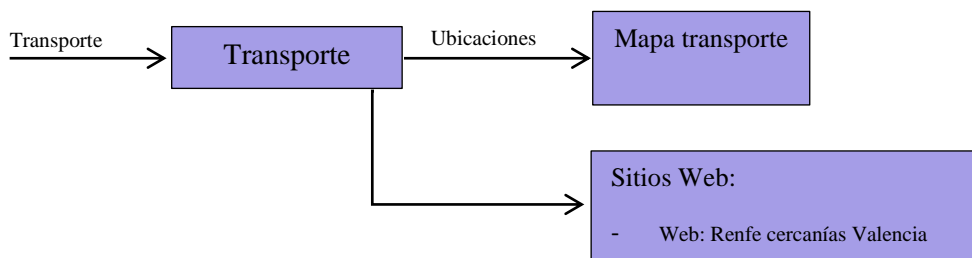
8.3.9. Flujograma talleres



Flujograma 17. Talleres.

La *activity* Talleres presenta, también, una estructura sencilla. Al seleccionar esta opción en `MenuPrincipal`, el usuario accede a una *activity* informativa, Talleres, que permite el acceso a una *activity* de Google Maps, `MapaTalleres`, dónde, mediante *markers*, se indica la posición de los talleres de la localidad junto a sus teléfonos.

8.3.10. Flujograma transporte

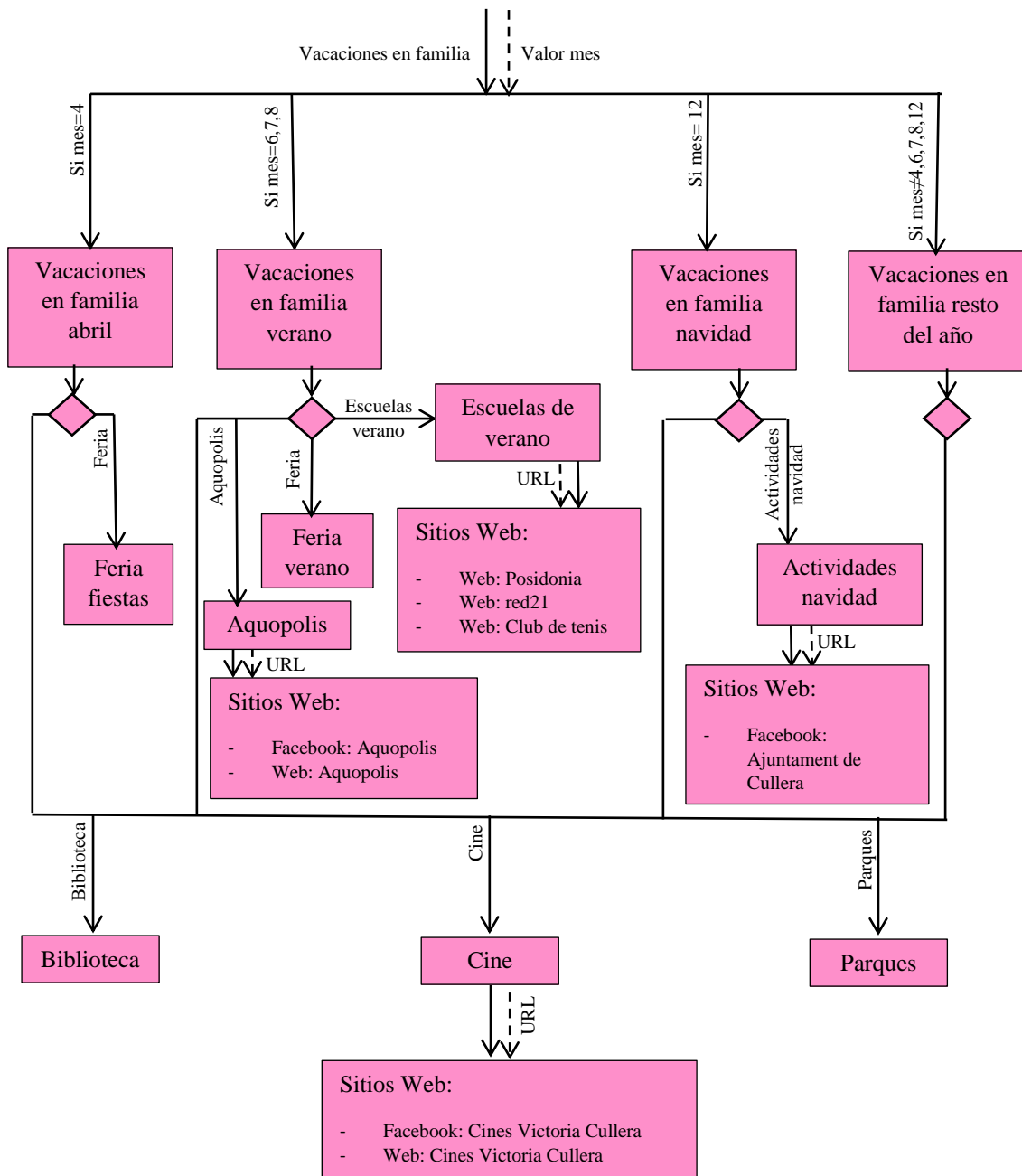


Flujograma 18. Transporte.

La estructura de la *activity* Transporte se basa en una *activity* informativa, Transporte, que permite el acceso a una *activity* de Google Maps, `MapaTransporte`, dónde, mediante *markers*, se indica la posición de las paradas de diferentes transportes.

Por otro lado, también tiene acceso a la *activity* SitiosWeb, que permite al usuario el acceso a la página web de Renfe.

8.3.11. Flujograma vacaciones en familia



Flujograma 19. Vacaciones en familia.

Este último flujograma expone la estructura de la app al escoger en MenuPrincipal la opción de consultar información acerca de vacaciones en familia. De esta manera, según el mes consultado, el usuario puede ser enviado a 4 *activities* diferentes que actúan como menús.

Todas las *activities* a las que se da acceso desde cada uno de los menús tienen carácter informativo. Además, algunas de estas, permiten al usuario acceder a la web o las redes sociales correspondientes, para ello se envían las URL a la *activity* *SitiosWeb* que se encarga de abrir estas páginas.



9. Conclusiones y propuesta de trabajo futuro

9.1. Conclusiones

Tras llegar a la finalización de la programación de la aplicación y de haberla explicado con detalle durante la elaboración de la memoria, puedo afirmar que he conseguido obtener una aplicación que facilita y mejora la experiencia del turismo en Cullera.

He conseguido reunir aquellos aspectos positivos que ya se utilizaban en aplicaciones similares y añadir nuevos aspectos que no habían sido contemplados y que resultan provechosos para los usuarios.

Por otro lado, respecto al diseño, he conseguido una interfaz atractiva, que llama la atención del usuario, y a su vez sencilla e intuitiva, que posibilita una buena experiencia durante su uso y facilita el acceso a la información.

La aplicación es una herramienta que ofrece información útil sobre todos los ámbitos de interés para un visitante de la localidad en un mismo soporte, eliminando la necesidad de consultar diferentes fuentes para conseguir la información que se busca.

Por todo lo anterior, he cumplido los objetivos que me había propuesto al inicio del proyecto y, además, este proceso me ha permitido aprender durante su realización.

9.2. Propuesta de trabajo futuro

Respecto a usos futuros, Disfruta Cullera podría ser utilizada con diferentes fines. Para ello debería ser completada a nivel profesional con más información y traducida en su totalidad para permitir su distribución entre clientes internacionales.

Un posible fin sería la adopción de esta como aplicación oficial para el turismo en Cullera siendo gestionada por el ayuntamiento, pero este ya dispone de una aplicación de las mismas características.

Otro uso que se le podría dar sería desde un organizador turístico, agencia de viajes o similar. Este podría ofrecerla a sus clientes como un servicio adicional al visitar el municipio, a su vez, se podrían ofertar los servicios que más les interesara ofrecer a los clientes.

En ambos casos la base de datos a la que está vinculada la aplicación permitiría analizar cuáles son los principales atractivos para los usuarios y, de esta manera, intentar fomentarlos o aprovecharlos para ofrecer a los turistas actividades de sus gustos.

10. Bibliografía

- [1] “Informe Ditrendia Mobile en España y en el Mundo 2017”, Ditrendia, España, 2018.
- [2] “Diseño de apps, su importancia y los mejores ejemplos” *Yeeply*, 2016. [En línea]. Disponible en: <https://www.yeeply.com/blog/disenodeapps-ejemplos/>. [Accedido: 12 mayo 2019]
- [3] “Diseño de apps: usabilidad y experiencia del usuario móvil” *Yeeply*, 2018. [En línea]. Disponible en: <https://www.yeeply.com/blog/disenodeaplicaciones-usabilidad-y-experiencia-de-usuario/>. [Accedido: 12-may-2019]
- [4] “Cuota de mercado de pedidos de smartphones a nivel mundial por sistema operativo entre 2014 y 2020”, *Statista*, 2016. [En línea]. Disponible en: <https://es.statista.com/estadisticas/600731/cuota-de-mercado-de-sistemas-operativos-para-smartphones-por-pedidos-2020/>. [Accedido: 14-may-2019].
- [5] “Curso de programación Android desde cero”, *YouTube*, 2017. [En línea]. Disponible en: <https://www.youtube.com/playlist?list=PLYvsggKtwbLX06iMtXnRGX5lyjiiMaT2y>. [Accedido: 05-nov-2018].
- [6] “Benidorm Guía turística con mapa - Aplicaciones en Google Play”, *GooglePlay*, 2019. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.minube.guides.benidorm>. [Accedido: 03-feb-2019].
- [7] “Cullera Turismo - Aplicaciones en Google Play”, *GooglePlay*, 2019. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.goodbarber.culleraturismo>. [Accedido: 03-feb-2019].
- [8] “GANDIA - Aplicaciones en Google Play”, *GooglePlay*, 2019. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.mirapordonde.gandia>. [Accedido: 03-feb-2019].
- [9] “Xàbia/Jávea - Aplicaciones en Google Play”, *GooglePlay*, 2019. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.turismoXabia>. [Accedido: 03-feb-2019].
- [10] “360 Movic - Diseño y desarrollo web / apps”, 2019. [En línea]. Disponible en: <https://www.360movic.com/>. [Accedido: 20-may-2019].
- [11] “minube: viajes, actividades, opiniones de qué ver y dónde dormir”, 2019. [En línea]. Disponible en: <https://www.minube.com/>. [Accedido: 20-may-2019].
- [12] “IdeaMixta | Agencia de Comunicación y Marketing - Crece tu negocio”, 2019. [En línea]. Disponible en: <http://www.ideamixta.com/>. [Accedido: 20-may-2019].
- [13] “Móvil y tercera edad (I): cifras y letras”, *Mobile World Capital Barcelona*, 07-ago-2013. [En línea]. Disponible en: <https://mobileworldcapital.com/es/2013/08/07/127/>. [Accedido: 23-may-2019].
- [14] “Conoce Android Studio”, *Android Developers*, 2019. [En línea]. Disponible en: <https://developer.android.com/studio/intro>. [Accedido: 25-may-2019].

- [15] “Android”, *Wikipedia, la enciclopedia libre*, 2019. [En línea].
Disponible en: <https://es.wikipedia.org/w/index.php?title=Android&oldid=116975416>
[Accedido: 25-may-2019].
- [16] “Firebase, qué es y para qué sirve la plataforma de Google”, *Blog de IEBSchool*,
14-oct-2016. [En línea]. Disponible en: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>. [Accedido: 25-may-2019].
- [17] “Lenguajes de programación que debes saber si quieres ser desarrollador Android”.
Yeeply, 2018. [En línea]. Disponible en: <https://www.yeeply.com/blog/lenguajes-basicos-desarrollador-android/>. [Accedido: 25-may-2019].
- [18] J. Dimas Luján “¿Qué es una Activity?”, *DesarrolloWeb.com*, 03-mar-2014. [En línea].
Disponible en: <https://desarrolloweb.com/articulos/android-que-es-una-activity-o-actividad.html>. [Accedido: 26-may-2019].
- [19] “Máster en Desarrollo de Aplicaciones Android - Uso de ConstraintLayout”, *Android Curso*, 2019. [En línea]. Disponible en:
<http://www.androidcurso.com/index.php/recursos/32-unidad-2-diseno-de-la-interfaz-de-usuario-vistas-y-layouts/881-uso-de-constraintlayout>. [Accedido: 26-may-2019].
- [20] A. Ardions “Qué tipos de layouts existen en Android Studio”, *AndroidStudioFaqs*,
09-abr-2015. [En línea]. Disponible en: <https://androidstudiofaqs.com/conceptos/que-tipos-de-layouts-existen-en-android-studio-androidstudiofaqs-te-lo-cuenta>.
[Accedido: 26-may-2019].
- [21] “Manifiesto de la app”, *Android Developers*, 2019. [En línea].
Disponible en: <https://developer.android.com/guide/topics/manifest/manifest-intro>.
[Accedido: 26-may-2019].
- [22] “Colocar Google Maps en tu aplicación de Android - Tutorial”, YouTube, 2016.
[En línea]. Disponible en: <https://www.youtube.com/watch?v=nybRpw2H7gg>.
[Accedido: 30-mar-2019].
- [23] “Android Studio - #1 Conectando con Firebase”, YouTube, 2017. [En línea]. Disponible
en: <https://www.youtube.com/watch?v=K8126eevcho>. [Accedido: 6-abr-2019].
- [24] “Java SE Development Kit 8 - Descargas”, *Oracle*, 2018. [En línea]. Disponible en:
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. [Accedido: 03-nov-2018].
- [25] “Download Android Studio and SDK tools”, *Android Developers*, 2018. [En línea].
Disponible en: <https://developer.android.com/studio>. [Accedido: 03-nov-2018].
- [26] “Wikiloc - Rutas del Mundo”, 2019. [En línea].
Disponible en: <https://es.wikiloc.com/>. [Accedido: 4-abr-2019].