

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN GESTIÓN DE LA INFORMACIÓN

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

CURSO ACADÉMICO 2016 - 2017



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

MUGI

**ANÁLISIS DE LAS FUNCIONES NECESARIAS
EN LA CREACIÓN DE UNA PLANTILLA PARA
UNA APLICACIÓN MÓVIL DE
AYUNTAMIENTO**

Presentado por:
SERGIO COMES MARÍ

Dirigido por:
NURIA LLORET ROMERO

VALENCIA, JULIO 2017

Índice de contenidos

1.	INTRODUCCIÓN	6
1.1.	JUSTIFICACIÓN	6
1.1.3.	EJEMPLOS DE AYUNTAMIENTOS	7
1.2.	OBJETIVOS	12
1.3.	ESTRUCTURA.....	13
2.	HERRAMIENTAS Y MÉTODOS	15
2.1.	PYTHON.....	15
2.2.	SCRAPING	15
2.3.	LIBRERÍAS DE PYTHON Y GOOGLE	16
3.	EXTRACCIÓN DE DATOS.....	18
3.1.	ACCESO A LA TIENDA DE GOOGLE PLAY	23
3.2.	CREACIÓN HOJA DE EXCEL.....	24
3.3.	CÓDIGO DE CONTROL DE SEGURIDAD.....	26
3.4.	OBTENCIÓN DE LA INFORMACIÓN DE LA APLICACIÓN.....	29
3.4.1.	PROBLEMAS CON LAS APLICACIONES SIN CORREO	35
3.4.2.	ARREGLAR LOS CORREOS ADECUADAMENTE	35
3.5.	EXPORTACIÓN EN EXCEL	36
4.	FILTRADO DE APLICACIONES.....	38
5.	ANÁLISIS E IDENTIFICACIÓN DE FUNCIONALIDADES	41
5.1.	APLICACIÓN DEL AYUNTAMIENTO DE CARTAGENA.....	41
5.2.	APLICACIÓN DEL AYUNTAMIENTO DE MIERES	45
5.2.1.	FUNCIONES ESENCIALES DE UN AYUNTAMIENTO	48

5.3. APLICACIÓN DEL AYUNTAMIENTO DE VALENCIA.....	51
6. SELECCIÓN FINAL DE FUNCIONES.....	59
7. RESULTADOS	65
8. MEJORAS PROPUESTAS POR DESARROLLAR.....	68
9. CONCLUSIONES.....	69
10. BIBLIOGRAFÍA.....	71
ANEXO I – CÓDIGO DE EXTRACCIÓN DE LA INFORMACIÓN.....	74
ANEXO II – APLICACIÓN DE CARTAGENA.....	76
ANEXO III – APLICACIÓN DE MIERES	77
ANEXO III – APLICACIÓN DE VALENCIA.....	78

Índice de ilustraciones

Ilustración 1. Tiempo medio usado en cada dispositivo.....	7
Ilustración 2. Aplicación Barcelona: Barcelona Guía Turística.....	9
Ilustración 3. Aplicación Madrid: Guía Bienvenidos a Madrid.	10
Ilustración 4. Aplicación Sevilla: Sevilla.....	12
Ilustración 5. Google Play Store con la palabra clave.....	18
Ilustración 6. Aplicación Zaragoza Rutas (1).....	19
Ilustración 7. Aplicación Zaragoza Rutas (2).....	20
Ilustración 8. Aplicación Zaragoza Rutas (3).....	21
Ilustración 9. Bloque de código del desarrollador de la aplicación.....	22
Ilustración 10. Código de extracción de datos (1): importación de librerías.	24
Ilustración 11. Código de extracción de datos (2): Creación de hoja de Excel.....	26
Ilustración 12. Código de extracción de datos (3): Control de seguridad (3.1).	28
Ilustración 13. Código de extracción de datos (3): Control de seguridad (3.2).	28
Ilustración 14. Código de extracción de datos (3): Control de seguridad (3.3).	29
Ilustración 15. Código de extracción de datos (3): Control de seguridad (3.4).	29
Ilustración 16. Código de extracción de datos (3): Control de seguridad (3.5).	29
Ilustración 17. Código de extracción de datos (4): Obtención de información de la aplicación (4.1).....	30
Ilustración 18. Código de extracción de datos (4): Obtención de información de la aplicación (4.2).....	31
Ilustración 19. Código de extracción de datos (4): Obtención de información de la aplicación (4.3).....	32
Ilustración 20. Código de extracción de datos (4). Obtención de información de la aplicación (4.4).....	32

Ilustración 21. Código de extracción de datos (4). Obtención de información de la aplicación (4.5).....	32
Ilustración 22. Código de extracción de datos (4). Obtención de información de la aplicación (4.6).....	33
Ilustración 23. Código de extracción de datos (4). Obtención de información de la aplicación (4.7).....	33
Ilustración 24. Información adicional: Datos dentro de la variable "correo".	34
Ilustración 25. Código de extracción de datos (4). Obtención de información de la aplicación (4.8).....	34
Ilustración 26. Código de extracción de datos (4). Obtención de información de la aplicación (4.9).....	35
Ilustración 27. Código de extracción de datos (4). Obtención de información de la aplicación (4.10).....	36
Ilustración 28. Exportación en Excel de la información.....	36
Ilustración 29. Aplicaciones descartadas.....	39
Ilustración 30. Aplicaciones de ayuntamiento válidas.....	40
Ilustración 31. Aplicación de Cartagena: Menú de inicio.....	42
Ilustración 32. Aplicación de Cartagena: Menú de actualidad.....	43
Ilustración 33. Aplicación de Cartagena: Menú de turismo.....	44
Ilustración 34. Aplicación de Cartagena: Menú de información general.	45
Ilustración 35. Aplicación de Mieres: Menú de inicio.	46
Ilustración 36. Aplicación de Mieres: Menú de "Más" (1).	47
Ilustración 37. Aplicación de Mieres: Menú de "Más" (2) - Conoce Mieres.....	48
Ilustración 38. Aplicación de Mieres: Menú de "Más" (3) - Servicios municipales.....	50
Ilustración 39. Aplicación de Valencia: Menú de funciones.....	51
Ilustración 40. Aplicación de Valencia: Menú de servicios cercanos.	53

Ilustración 41. Aplicación de Valencia: Menú de trámites y gestiones.	54
Ilustración 42. Aplicación de Valencia: Información municipal.	55
Ilustración 43. Aplicación de Valencia: Listado de avisos.	56
Ilustración 44. Aplicación de Valencia: Menú de cultura y fiestas populares.	57
Ilustración 45. Aplicación de Valencia: Menú de aplicaciones.	58
Ilustración 46. Funciones almacenadas en Excel.	59
Ilustración 47. Comparación de las funciones en las diversas aplicaciones analizadas (1).	60
Ilustración 48. Comparación de las funciones en las diversas aplicaciones analizadas (2).	61
Ilustración 49. Funciones escogidas para la plantilla (1).	63
Ilustración 50. Funciones escogidas para la plantilla (2).	64

1. INTRODUCCIÓN

El siguiente proyecto surge de la necesidad que tienen los ayuntamientos de España en actualizarse para poder llegar a todos los ciudadanos, visitantes o turistas. Además de servir como fuentes de información, las aplicaciones suponen una nueva oportunidad de explotación comercial y turística de la ciudad.

El proyecto consiste en crear una estructura estándar que sirva de punto de partida en la creación de una aplicación móvil para ayuntamientos.

1.1. JUSTIFICACIÓN

Las personas y la tecnología van de la mano hoy en día, desde que nos levantamos hasta que nos vamos a dormir, tenemos una constante influencia de tecnología a lo largo de la jornada. Según el informe *“How People Use Their Devices”* de Google, el 80% de los usuarios utilizan un teléfono móvil frente al 67% que usa un ordenador y el 16% que opta por la Tablet, de hecho, para algunos usuarios solo existe el móvil. Pero, ¿cuánto tiempo invertimos en los distintos dispositivos?

Una persona suele pasar 170 minutos conectado a su Smartphone, casi 3 horas diarias; mientras que los usuarios de ordenador invierten 120 minutos diarios en este dispositivo. Por último, el tiempo medio del uso de la Tablet es de 75 minutos al día.

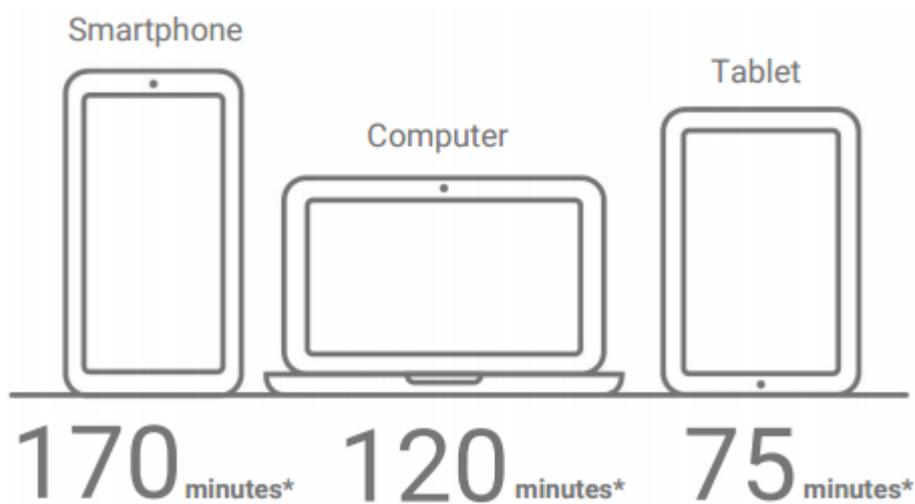


Ilustración 1. Tiempo medio usado en cada dispositivo.

Conociendo estos datos, no es de extrañar que las empresas se adapten a las nuevas demandas de la sociedad cuya principal característica es la interconectividad, apostando por el m-commerce (comercio electrónico usando el teléfono móvil) o realizando estrategias de social media.

Pero además de las empresas, los gobiernos y ayuntamientos decidieron que también debían subirse al tren tecnológico y actualmente, en España, muchos de ellos cuentan con su propia aplicación móvil, dentro de la tienda de Google Play.

1.1.3. EJEMPLOS DE AYUNTAMIENTOS

Algunos de los ayuntamientos de las ciudades más importantes de España ya cuentan con una o varias aplicaciones de móvil. A continuación, se mostrarán aplicaciones de Barcelona, Madrid y Sevilla.

Barcelona:

Nombre de la aplicación: Barcelona Guía Turística

Valoración: 4 estrellas.

Descripción:

Es una guía de los sitios que se puede visitar en Barcelona. Cuenta con un mapa interactivo donde el usuario puede navegar por la ciudad de Barcelona y crear sus propias rutas. Además, permite al usuario buscar por un interés concreto:

- Hotel.
- Comida.
- Local nocturno.
- Arte y entretenimiento.
- Transporte público.
- Tienda y servicio.
- Aire libre y recreación.

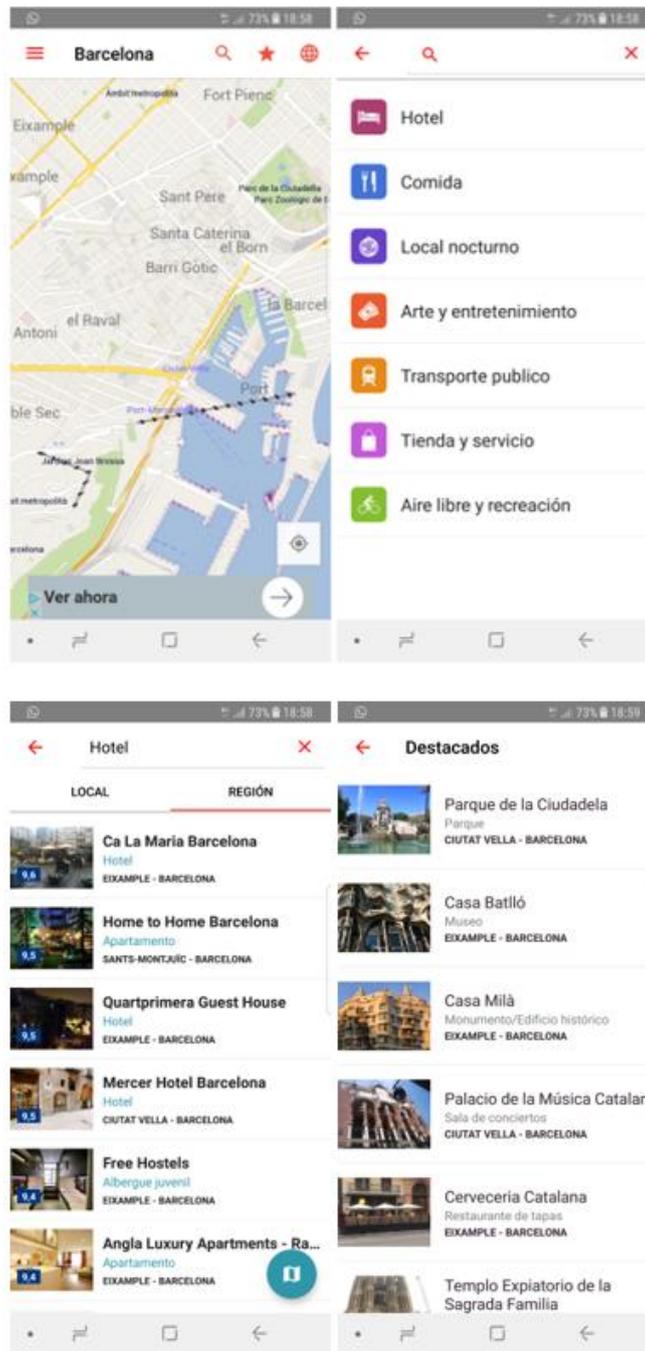


Ilustración 2. Aplicación Barcelona: Barcelona Guía Turística.

Madrid:

Nombre de la aplicación: Guía Bienvenidos a Madrid.

Valoración: 4 estrellas.

Descripción:

Es una guía que ofrece todo lo necesario para disfrutar de la ciudad de Madrid. Contiene información esencial para organizar el viaje y visitar los lugares más importantes, además de ofrecer consejos turísticos y agenda completa y actualizada.

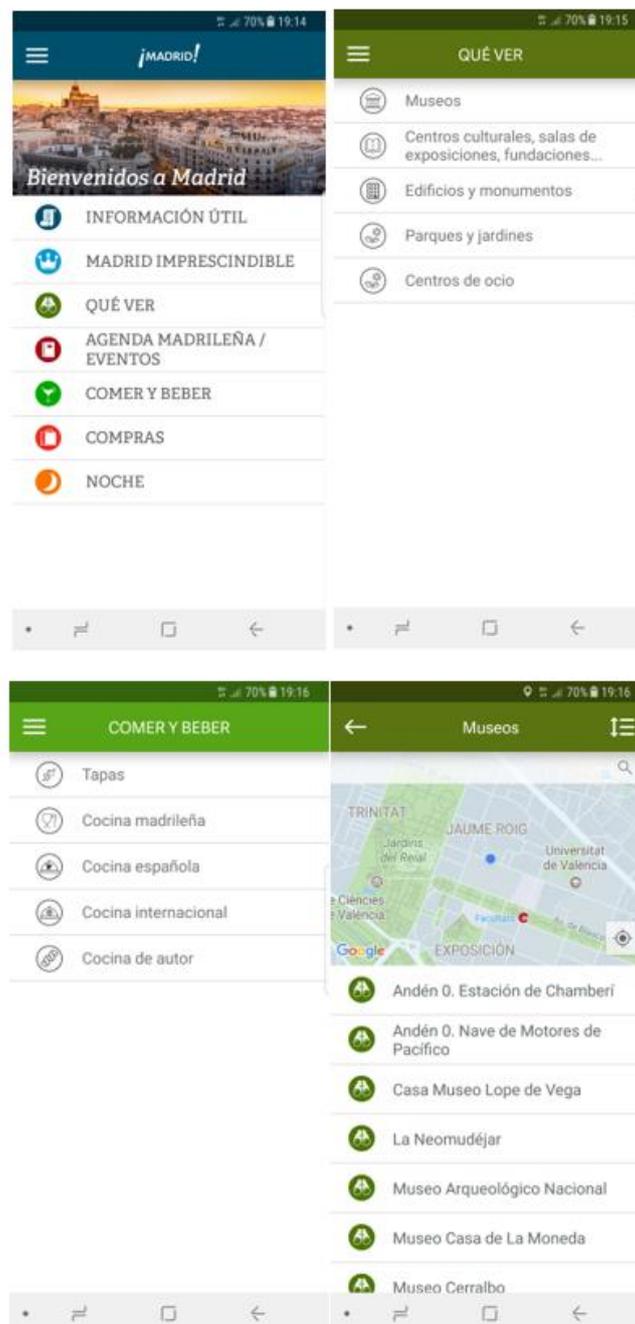


Ilustración 3. Aplicación Madrid: Guía Bienvenidos a Madrid.

Sevilla:

Nombre de la aplicación: Sevilla.

Valoración: 3,9 estrellas.

Descripción:

La aplicación oficial de turismo de Sevilla contiene las funcionalidades necesarias para facilitar la estancia del visitante. Incorpora un buscador de experiencias y actividades turísticas en la ciudad, con posibilidad de reserva. Además, brinda información detallada sobre los monumentos más importantes. Contiene diversa información:

- Puntos de interés.
- Gastronomía.
- Turismo familiar.
- Agenda de eventos.
- Guía de compras.
- Rutas temáticas.
- Servicio localizador.

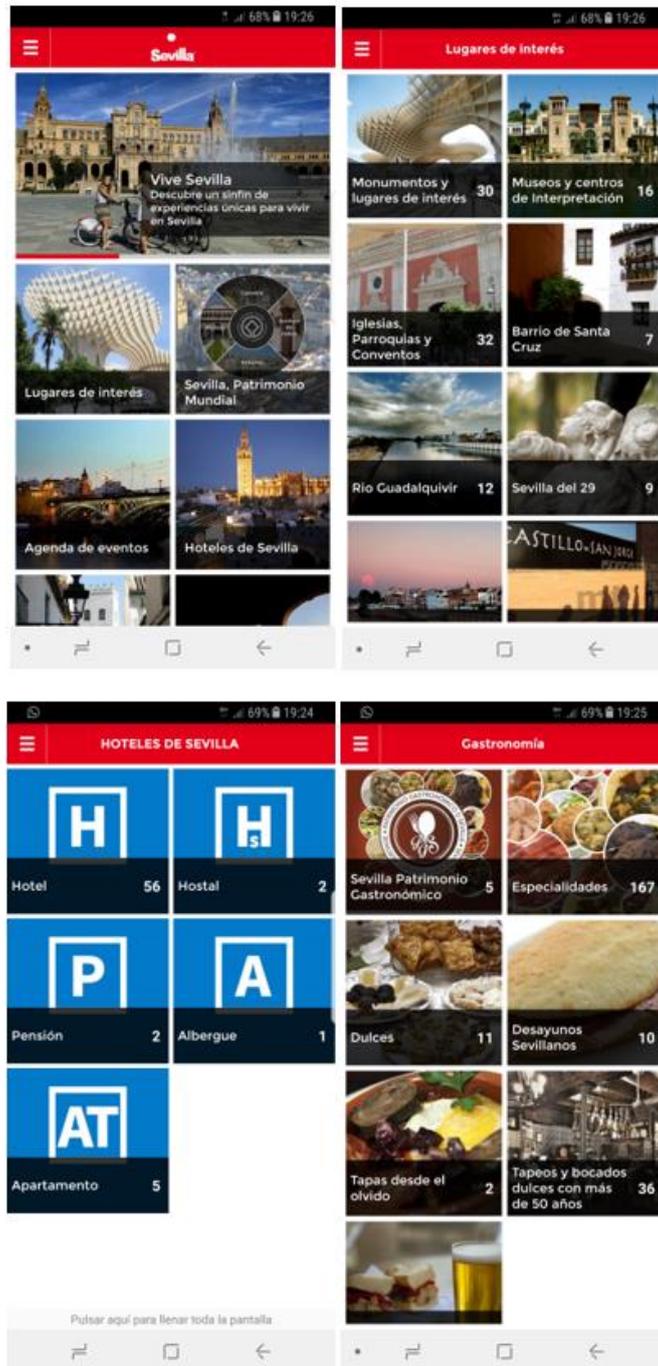


Ilustración 4. Aplicación Sevilla: Sevilla.

1.2. OBJETIVOS

El objetivo principal del proyecto consiste en realizar una estructura que sirva como guía para una futura aplicación móvil de ayuntamiento. Para ellos se ha fijado una serie de objetivos específicos:

- Recuperar e investigar las aplicaciones móviles a través de la herramienta de Google Play Store (<https://play.google.com/store/apps?hl=es>).
- Analizar las funciones de dichas aplicaciones y organizarlas de forma que recojan la mayor información posible en la plantilla.

1.3. ESTRUCTURA

Teniendo en cuenta los objetivos específicos, se ha dividido el trabajo en dos partes. El primer apartado se basa en un código escrito, manualmente, en el lenguaje de programación Python. Dicho programa trabaja con ciertas librerías de Google para la extracción de información de su herramienta Google Play Store.

A través de código, se crea una hoja de Excel donde guardar los datos relevantes que interesa exportar. Después, a través del uso de la técnica de Scraping, se accede al código de la página web de la tienda y se obtiene el título de la aplicación junto con el número de descargas, la última fecha de actualización, la valoración y la URL.

El segundo apartado se trata de filtrar las aplicaciones de ayuntamiento, que es lo que realmente interesa, ya que se pueden haber colado otras aplicaciones que no resultan de utilidad, como juegos de móvil, aplicaciones de parking, etc.

Como trabajo extra, también se obtienen los correos electrónicos de las aplicaciones para un futuro contacto comercial.

Una vez eliminado lo restante, se accede a algunas de las aplicaciones más descargadas (que indicará que son las más utilizadas), así como las de ciudades importantes como Madrid, Barcelona, Valencia o Sevilla. No obstante, también se mirarán algunas poco descargadas para ver el contraste entre una aplicación u otra, y determinar que funciones son o no recomendables introducir.

Por último, debido a que no todas las aplicaciones tienen las mismas funciones, en un primer paso se tomará nota de todas aquellas necesarias para un ayuntamiento, aunque indican cosas similares; y en un segundo paso, se recopilarán las funciones comunes en una sola para mostrar el mayor contenido en el menor espacio posible, ya que el móvil tiene una pantalla reducida de visión.

2. HERRAMIENTAS Y MÉTODOS

Para la realización del siguiente proyecto se han empleado diversas herramientas, así como métodos necesarios para la extracción de datos, correspondientes al primer apartado.

2.1. PYTHON

Python es un lenguaje de programación multiparadigma, es decir, en vez de forzar a los programadores a adoptar un estilo particular de programación, soporta varios estilos como orientación a objetos, programación imperativa y programación funcional.

Tiene una licencia de código abierto compatible con la licencia de código abierto de GNU a partir de la versión 2.1.1 e incompatible con las versiones anteriores.

Python fue diseñado para ser leído con facilidad. El contenido de los bloques de código (bucles, funciones, etc.) es delimitado mediante espacios o tabuladores antes de cada línea de órdenes pertenecientes al bloque, de ese modo se diferencia de otros lenguajes de programación que tienen como costumbre declarar estos bloques mediante llaves ({}).

Otra novedad de Python es que las variables se definen de forma dinámica, lo que significa que no se tienen que especificar de antemano y puede tomar distintos valores en otro momento.

Las funciones se definen con la palabra clave “def”, seguida del nombre de la función y sus parámetros. El valor se devuelve con la instrucción “return”.

Python tiene una biblioteca estándar bastante grande. Los módulos de la biblioteca pueden mejorarse por módulos personalizados escritos tanto en C como en Python.

Este lenguaje de programación es perfecto para realizar este proyecto ya que dispone de las librerías necesarias para usar el método de “scraping” en la tienda de Google Play.

2.2. SCRAPING

Scraping o web scraping es una técnica utilizada para extraer información de los sitios web. Esta programación simula la navegación de un ser humano en la web. Sin embargo, el scraping se enfoca en la información de datos estructurados que pueden ser almacenados y analizados en

una base central, hoja de cálculo o alguna fuente de almacenamiento. En los últimos años, el scraping se ha convertido en una técnica muy utilizada dentro del sector de posicionamiento web gracias a su capacidad de generar grandes cantidades de datos para crear contenidos de calidad.

Gracias al scraping se podrá entrar en la tienda de Google Play y mediante el código implementado más adelante, obtener la información necesaria de todas las aplicaciones de ayuntamiento en un minuto. Si se realizará de forma manual esta tarea, se tendría que entrar en cada aplicación y copiar uno a uno los datos que se requieren, faena que llevaría por lo menos 4 o 5 horas aproximadamente.

2.3. LIBRERÍAS DE PYTHON Y GOOGLE

Python tiene varias librerías o módulos útiles que se han usado en este proyecto debido a su facilidad a la hora de obtener información y almacenarla. Las siguientes librerías que se han usado son:

- El módulo **sys** provee acceso a funciones y objetos mantenidos por el intérprete. En este caso, permitirá ejecutar la función del código a través de la consola de Linux permitiendo pasarle argumentos.
- La librería **requests** facilita usar métodos sencillos con estructura REST. Cuenta con los métodos “post”, “get”, “put”, “delete”, “patch”, “head” y “options”, los cuales permiten la comunicación con un API RESTful sin problemas. En este caso, ayudará a comunicarse con el código html de cada aplicación para poder extraer la información que no tiene en cuenta la API de Google.
- El módulo **xlwt** sirve para trabajar con hojas de Excel desde Python de forma bastante sencilla. Concretamente, el módulo xlwt sirve para escribir (el único que se usará en este proyecto), y cuenta con otros dos módulos más, xlrd y xlutils, para poder modificar o copiar un Excel, respectivamente. Este módulo se usará para organizar la información almacenada en el Excel y que sea más legible a la hora de manejar los datos.
- El módulo **pprint** proporciona la capacidad de “imprimir” estructuras de datos Python arbitrarias en una forma que se pueda utilizar como entrada para el intérprete. Si las estructuras incluyen objetos que no son tipos de Python, la representación no puede

ser cargable. Pprint guarda objetos en una sola línea si se puede, y los rompe en varias líneas si no encajan dentro de la anchura permitida.

- La librería **BeautifulSoup** proporciona métodos simples para navegar y extraer la información que se necesite. En este caso se creará un objeto de la clase BeautifulSoup, pasándole el HTML de la aplicación concreta. Gracias a ese objeto, se puede identificar el tipo de etiqueta que se quiere obtener.
- La librería **GooglePlayAPI** proporciona, al igual que BeautifulSoup, métodos para extraer la información de las aplicaciones que se muestra en la tienda Google Play. Además, posee funciones (que se utilizarán) para hacer búsquedas por palabras clave, útil y de gran necesidad para este proyecto. En esta librería, también se encuentra incluido el módulo **helpers**, que proporciona más métodos un poco más específicos.

3. EXTRACCIÓN DE DATOS

En este apartado se resume la parte del código realizada para la extracción de los datos de la tienda de GooglePlay. Se irá explicando paso a paso para que se entienda de la forma más sencilla posible.

El primer paso será entrar en la web de Google Play Store e identificar el tipo de HTML que tiene, ya que para utilizar el scraping se debe conocer la estructura de la página web (concretamente en que bloques y etiquetas está almacenada la información, por ejemplo, dentro de un bloque con la etiqueta <div>).

Para ello se accede a través de la URL de la tienda (<https://play.google.com/store/apps?hl=es>) y en el buscador se escribe la palabra clave ayuntamiento (es el tipo que realmente interesa identificar).

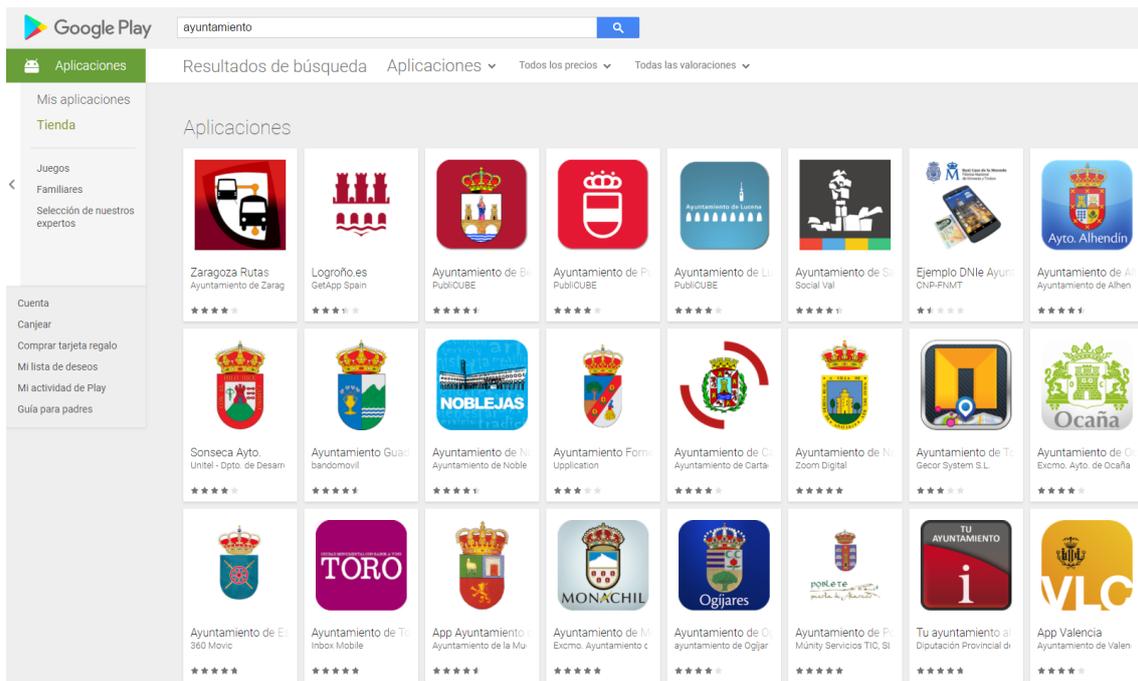


Ilustración 5. Google Play Store con la palabra clave.

Gracias a la librería de GooglePlayAPI, los métodos que proporciona permiten acceder a cada una de las aplicaciones que muestra la web (en total hay unas 250) mediante una palabra clave, por lo que realmente interesa saber la estructura del código de una aplicación. Con saber la estructura del código de una sola aplicación será suficiente ya que todas comparten la misma.

Accediendo a la primera aplicación (por elegir una cualquiera) se puede ver el título de la aplicación, un breve “slide” de imágenes junto con una breve descripción de ella. Más abajo se encuentran los comentarios y el apartado “Novedades” y finalmente la información adicional.



Zaragoza Rutas
Ayuntamiento de Zaragoza · Mapas y navegación · ★★★★★ 422

3 PEGI 3

Esta aplicación es compatible con todos tus dispositivos.

Añadir a la lista de deseos **Instalar**

Zaragoza RUTAS
Cómo moverse en transporte público

Líneas de transporte · Rutas · Favoritos · Tiempos de llegada · Paradas próximas · Créditos

(325) CONDES ARAGÓN 16

Línea actual

- 35 3 minutos VIA HISPANIDAD
- 35 6 minutos VIA HISPANIDAD

Otras líneas de la parada

- 42 1 minuto LA PAZ
- 42 7 minutos LA PAZ
- 53 10 minutos MIRALBUENO
- 53 14 minutos MIRALBUENO

TIEMPOS DE LLEGADA

Escanea el código QR o introduce el número de poste para conocer los tiempos de llegada:

- Escanear código QR
- Número de poste: 445

Autobús · Tranvía · Autobús barrios rurales

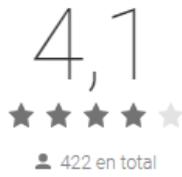
Aplicación oficial del Ayuntamiento de Zaragoza que permite obtener **la mejor ruta para desplazarse por la ciudad en transporte público**.

Permite a la ciudadanía planificar sus desplazamientos en tiempo real por Zaragoza mediante el cálculo de itinerarios personalizados, teniendo en cuenta sus preferencias respecto al uso del tranvía y/o autobús (incluidas las líneas de barrios rurales), momento exacto y día de la semana, etc. y

MÁS INFORMACIÓN

Ilustración 6. Aplicación Zaragoza Rutas (1).

OPINIONES



Beatriz Gálvez ★★★★★
De las mejores que he tenido en cuanto a que funciona...pero o no es precisa o los buses



Carmen Po ★★★★★
Muy útil. Se puede buscar por paradas mas cercanas, por líneas, por postes. Te muestra



Marisa Vicente ★★★★★
Muy útil. Puedes buscar por N° de ruta o las paradas más cercanas tanto buses como



Ascensión Pérez Alonso ★★★★★
Es muy precisa. Me encanta, es la mejor aplicación que existe en el tema de transporte



NOVEDADES

¡NOVEDADES!

- Informamos de los DESVÍOS en tiempo real en las paradas
- Ya puedes conocer la ACCESIBILIDAD de los autobuses urbanos pendientes de llegar a tu parada (un icono junto a los minutos para la llegada indica si dispone de rampa adaptada para facilitar el acceso)
- Conoce al instante las INCIDENCIAS del servicio. Una alerta al iniciar la app te mostrará las incidencias en tiempo real.

Ilustración 7. Aplicación Zaragoza Rutas (2).

INFORMACIÓN ADICIONAL		
Actualizado 22 de junio de 2017	Descargas 50.000 - 100.000	Versión actual 1.4.2
Requiere Android 2.1 y versiones superiores	Clasificación del contenido PEGI 3 Más información	Permisos Ver detalles
Informe Marcar como inapropiado	Ofrecida por Ayuntamiento de Zaragoza	
Desarrollador Visitar sitio web Correo electrónico: internet@zaragoza.es Política de Privacidad		

Ilustración 8. Aplicación Zaragoza Rutas (3).

La información que se requiere está dispersa por toda la página web, el título está al principio; el número de estrellas está en el apartado de “Opiniones”; el correo electrónico, el número de descargas y la última actualización están en el apartado de “Información adicional”; y la URL de la aplicación no se encuentra almacenada por ninguna parte.

No obstante, que la información no esté junta no es ningún problema ya que, la librería de GooglePlayAPI guarda esa información en variables predefinidas, por ejemplo, el título de la aplicación lo guarda en la variable “title”, la URL la guarda en la variable “shareURL”, todos los datos que se requieren GooglePlayAPI lo proporciona excepto el correo electrónico, por ese motivo, se usa el método de scraping. Sabiendo la información que se quiere obtener, hay que ver en que bloque y etiqueta se almacena para poder extraerla.

Al hacer “click” derecho en la web, aparece una opción de “Ver código fuente de la página” que mostrará el código todo junto sin espacios, casi ilegible para aquel que no sea programador o que no posea conocimientos de HTML.

Para esclarecer el asunto, recordar que la información que se necesita es el correo electrónico. Todo lo que aparece en la página web está escrito de alguna forma u otra en el código, por lo que se puede hacer un barrido de búsqueda con la palabra clave “Correo electrónico” mediante la orden Ctrl+F. Al realizar la búsqueda se encuentra una única coincidencia.

En HTML, el contenido está almacenado por etiquetas: el texto suele estar entre las etiquetas `<p></p>`; los enlaces entre las etiquetas `<a>`; los bloques entre las etiquetas `<div></div>`. Existen numerosas etiquetas, pero, básicamente, las tres mencionadas son las únicas que se van a utilizar.

Para explicar mejor el bloque donde se encuentra el “Correo electrónico” se ha copiado el código a un editor de texto de forma que sea más sencillo de entender.

```
<div class="meta-info meta-info-wide">
  <div class="title">Desarrollador</div>
  <div class="content contains-text-link">
    <a class="dev-link"
      href="https://www.google.com/url?q=http://www.zaragoza.es/ciudad/viapublica/movil
      rel="nofollow" target="_blank">
      Visitar sitio web
    </a>
    <a class="dev-link" href="mailto:internet@zaragoza.es"
      rel="nofollow" target="_blank">
      Correo electrónico: internet@zaragoza.es
    </a>
    <a class="dev-link"
      href="https://www.google.com/url?q=https://www.zaragoza.es/sede/portal/politica-r
      rel="nofollow" target="_blank">
      Política de Privacidad
    </a>
  </div>
</div>
```

Ilustración 9. Bloque de código del desarrollador de la aplicación.

Como se ve en la imagen, el “Correo electrónico” está almacenado dentro de un bloque entre etiquetas `<div></div>` cuyo nombre de la clase se llama “content contains-text-link”. Con esta información se puede decir, mediante scraping, que busque el bloque con ese nombre de clase, el problema es que hay varios con ese nombre, por lo que no resultaría muy eficiente y se perdería mucho tiempo.

Por otra parte, se puede observar que, además del bloque, en un nivel inferior, el “Correo electrónico” también se encuentra dentro de unas etiquetas `<a>` cuyo nombre de clase es “dev-link”. En este caso, se ha comprobado de que solo existen 3 enlaces con ese tipo de nombre, que son los del bloque del Desarrollador (los 3 enlaces que se muestran en la imagen), así pues se ha reducido el número de enlaces.

Ahora solo interesa conseguir de los 3 enlaces el que contiene el “Correo electrónico”. Después de pensarlo mucho, se llegó a la solución de utilizar la función “find()”. Esta función detecta si un subconjunto de palabras está dentro de otro conjunto, es decir, si un texto cualquiera está

dentro de otro texto. Como mediante el scraping obtenemos el valor de “href” de la etiqueta <a>, se deberá encontrar una palabra clave que distinga el “Correo electrónico” del resto de enlaces. Si se observa bien, la palabra “mailto” es el identificador perfecto, ya que solo aparece en el enlace del “Correo electrónico” y es bastante significativo.

Una vez se tiene los pasos a seguir, es hora de implementarlo todo en un código.

3.1. ACCESO A LA TIENDA DE GOOGLE PLAY

El primer paso será incorporar las librerías para que las detecte y se puedan utilizar los métodos y funciones de estos. Para ello hay que importarlas de la siguiente forma:

- `import requests`

La función “import” importa los módulos o librerías al código. También hay una variante de la declaración “import” que importa los nombres de un módulo directamente al espacio de nombres que hace la importación:

- `from bs4 import BeautifulSoup`

La diferencia entre una forma y otra es la manera de acceder a las funciones del módulo. Si no se pusiera “from bs4” se accedería a la función a través de:

- `bs4.BeautifulSoup.funcion()`

De la otra forma se ahorra el “bs4”, en general, se escribe más sencillo el código.

```

#!/usr/bin/python

# Do not remove
GOOGLE_LOGIN = GOOGLE_PASSWORD = AUTH_TOKEN = None

import sys
import requests
import xlwt
from pprint import pprint
from bs4 import BeautifulSoup

from config import *
from googleplay import GooglePlayAPI
from helpers import sizeof_fmt, print_header_line, print_result_line

```

Ilustración 10. Código de extracción de datos (1): importación de librerías.

Con las librerías importadas, ya se tiene acceso a todas las funciones de GooglePlayAPI, por lo tanto acceso a Google Play Store.

3.2. CREACIÓN HOJA DE EXCEL

Lo segundo que se ha hecho ha sido crear la hoja de Excel donde se almacenará la información, para ello se usa el módulo xlwt. Este módulo permite crear una hoja nueva, ponerle título, cambiar los estilos, añadir bordes... Para empezar, se crea una nueva hoja y se almacena en la variable “wb” a través de la siguiente instrucción:

- `wb = xlwt.Workbook()`

La función “Workbook()” es la responsable de crear una hoja nueva, cada vez que se quiera crear una nueva, se deberá llamar a esta función. El siguiente paso es añadirle un título a esa hoja:

- `ws = wb.add_sheet('A Test Sheet')`

A partir de la hoja creada, se llama a la función “add_sheet” que le asigna el nombre que se indique dentro de los paréntesis. Después se crean las columnas y se definen que longitud tendrá cada una:

- `ws.col(0).width = 13 * 625`
- `ws.col(1).width = 13 * 625`
- `ws.col(2).width = 13 * 625`
- `ws.col(3).width = 13 * 625`
- `ws.col(4).width = 13 * 625`
- `ws.col(5).width = 26 * 625`

Las columnas funcionan como si fuera una lista llamada “col”, la primera columna es la posición “0”, y la última columna es la posición “5”. Para darles tamaño, se entra en el campo “width” de cada columna y se añade la longitud. Se han creado 6 columnas correspondientes al nombre de la aplicación, correo electrónico, número de descargas, última actualización, número de estrellas y URL de la aplicación (la última columna, al ocupar la URL de la aplicación de ha creado más larga, para que no haya cortes). Una vez creadas se le añade el estilo a la primera fila, correspondiente al título de las columnas:

- `style = 'pattern: pattern solid, fore_colour yellow;'`
- `fmt = xlwt.Style.easyxf("""borders: left thick, right thick, top thick, bottom thick; pattern: pattern solid, fore_colour yellow;""")`

Los estilos van entre 3 comillas dobles, tanto el color como los bordes. Mediante la función “Style.easyxf” se le añade el estilo a la variable “fmt”, que más tarde se usará en cada columna. Por último, se le da nombres a las columnas para reconocer que información va en cada una:

- `ws.row(0).write(0, "NOMBRE DE LA APLICACION", fmt)`
- `ws.row(0).write(1, "CORREO ELECTRONICO", fmt)`
- `ws.row(0).write(2, "NUMERO DE DESCARGAS", fmt)`

- `ws.row(0).write(3,"ULTIMA ACTUALIZACION",fmt)`
- `ws.row(0).write(4,"NUMDE ESTRELLAS (MAX 5)",fmt)`
- `ws.row(0).write(5,"ENLACE DE LA APLICACION",fmt)`

En este trozo de código se está escribiendo en la primera línea a través de “`ws.row(0)`” gracias a la función “`write()`”. Dentro de la función se le añade la posición de la columna, el nombre que va a recibir esa columna, y los estilos.

```
#Excel
wb = xlwt.Workbook()
ws = wb.add_sheet('A Test Sheet')
ws.col(0).width = 13 * 625
ws.col(1).width = 13 * 625
ws.col(2).width = 13 * 625
ws.col(3).width = 13 * 625
ws.col(4).width = 13 * 625
ws.col(5).width = 26 * 625
style = 'pattern: pattern solid, fore_colour yellow;'
fmt = xlwt.Style.easyxf("""borders: left thick, right thick,
top thick, bottom thick;pattern: pattern solid, fore_colour yellow;""")
ws.row(0).write(0,"NOMBRE DE LA APLICACION",fmt)
ws.row(0).write(1,"CORREO ELECTRONICO",fmt)
ws.row(0).write(2,"NUMERO DE DESCARGAS",fmt)
ws.row(0).write(3,"ULTIMA ACTUALIZACION",fmt)
ws.row(0).write(4,"NUM DE ESTRELLAS (MAX 5)",fmt)
ws.row(0).write(5,"ENLACE DE LA APLICACION",fmt)
```

Ilustración 11. Código de extracción de datos (2): Creación de hoja de Excel.

3.3. CÓDIGO DE CONTROL DE SEGURIDAD

A la hora de ejecutar el código, se hará a través de consola con la instrucción:

- `python search.py ayuntamiento [250] [0]`

Esta instrucción corresponde con los siguientes campos:

- La primera palabra es la posición “0”, en ella se indica el programa que va a ejecutar el programa.
- La segunda palabra es la posición “1” e indica el código que se va a ejecutar. Como veis, hay que escribirlo con su terminación (en este caso es .py ya que está escrito en Python).
- La tercera palabra es la posición “2” e indica la palabra clave que se le añade en la web de Google Play para la búsqueda de aplicaciones. En este caso, se buscarán las aplicaciones de ayuntamiento.

A partir de aquí, es obligatorio estas 3 primeras posiciones, es decir, con escribir tan solo “python search.py ayuntamiento” el código funcionaría (sacaría solo la información de la primera aplicación y pararía), sin embargo, lo que realmente interesa es la información de todas las aplicaciones, para ello hay un campo extra.

- La cuarta palabra es la posición “3”, con ella se indica el número de aplicaciones que se quiere investigar. En el caso de poner un número demasiado alto, no habría ningún problema ya que cuando llega a la última aplicación pararía. En este proyecto se ha visto que hay 250 aplicaciones, por ese motivo se ha añadido el número exacto.

La última palabra corresponde a la posición “4”, pero en este proyecto no se va a utilizar. Este campo muestra información extra de la búsqueda en tiempo de ejecución de cada aplicación. Es un campo booleano, es decir, si se pone un “0” equivale a un “FALSE” por lo que no muestra nada; si se pone un “1” equivaldría a un “TRUE”, y mostraría la información extra.

Todo esto también hay que especificarlo dentro del código, así pues, procedemos a ello.

```

if (len(sys.argv) < 2):
    print "Usage: %s request [nb_results] [offset]" % sys.argv[0]
    print "Search for an app."
    print "If request contains a space, don't forget to surround it with \"\""
    sys.exit(0)

request = sys.argv[1]
nb_res = None
offset = None

if (len(sys.argv) >= 3):
    nb_res = int(sys.argv[2])

if (len(sys.argv) >= 4):
    offset = int(sys.argv[3])

api = GooglePlayAPI(ANDROID_ID)
api.login(GOOGLE_LOGIN, GOOGLE_PASSWORD, AUTH_TOKEN)

try:
    message = api.search(request, nb_res, offset)
except:
    print "Error: something went wrong. Maybe the nb_res you specified was too big?"
    sys.exit(1)

```

Ilustración 12. Código de extracción de datos (3): Control de seguridad (3.1).

Gracias a la librería “sys” se puede realizar estas comprobaciones por seguridad. Básicamente, se va a asegurar de que el número de parámetros que se le añade a la instrucción anterior para ejecutar el código sea el correcto, ni más ni menos.

El primer apartado indica que si el número de parámetros es menor a 2 muestre un error explicando porque falla. En este caso, le faltaría incluir la palabra clave “ayuntamiento” para realizar la búsqueda. De modo que, con este trozo de código se asegura que, si no hay como mínimo 3 parámetros en la instrucción, de error.

```

if (len(sys.argv) < 2):
    print "Usage: %s request [nb_results] [offset]" % sys.argv[0]
    print "Search for an app."
    print "If request contains a space, don't forget to surround it with \"\""
    sys.exit(0)

request = sys.argv[1]
nb_res = None
offset = None

```

Ilustración 13. Código de extracción de datos (3): Control de seguridad (3.2).

Después muestra la opción en el que hubiera 4 o 5 parámetros. En ese caso, coge el valor del “array” correspondiente a cada campo y se lo asigna a su variable identificadora.

```
if (len(sys.argv) >= 3):
    nb_res = int(sys.argv[2])

if (len(sys.argv) >= 4):
    offset = int(sys.argv[3])
```

Ilustración 14. Código de extracción de datos (3): Control de seguridad (3.3).

A continuación, detecta el usuario y la contraseña para poder acceder a la aplicación de Google Play Store, pero como se ha comprobado a lo largo de la realización del proyecto, no es necesario iniciar sesión ni registrarse para poder trabajar con ella. Debe ser, por si alguien quisiera entrar con su usuario y contraseña.

```
api = GooglePlayAPI(ANDROID_ID)
api.login(GOOGLE_LOGIN, GOOGLE_PASSWORD, AUTH_TOKEN)
```

Ilustración 15. Código de extracción de datos (3): Control de seguridad (3.4).

Por último, si todo ha ido bien, llama a la función “search” con los parámetros que se le han pasado desde un principio y guarda el valor en la variable “message”. En caso de que haya habido un error, pasa a la opción “except” y muestra un mensaje de error.

```
try:
    message = api.search(request, nb_res, offset)
except:
    print "Error: something went wrong. Maybe the nb_res you specified was too big?"
    sys.exit(1)
```

Ilustración 16. Código de extracción de datos (3): Control de seguridad (3.5).

3.4. OBTENCIÓN DE LA INFORMACIÓN DE LA APLICACIÓN

A partir de aquí viene, realmente, la parte más interesante de todo el código. Lo primero será crear un “array” para cada campo que se quiera almacenar, de ese modo los tendremos todos bien organizados, cada uno en su grupo. Para ello, se crean 6 “arrays” vacíos:

```
apps = []
mails = []
descargas = []
actualizaciones = []
ranking = []
enlaces = []
```

Ilustración 17. Código de extracción de datos (4): Obtención de información de la aplicación (4.1).

El siguiente paso es recorrer todas las aplicaciones de “ayuntamiento”. Para realizar esta tarea se usará un bucle “for” en el que iremos almacenando todos los datos de cada aplicación en su “array” correspondiente, sin embargo, con el “Correo electrónico” se tiene el problema tratado anteriormente, para ello, se plantea la solución del scraping.

Una vez guardados los campos que Google facilita a través de su librería, el campo de “Correo electrónico” es más complicado de obtener. Se deberá distinguir entre toda la información el texto del correo y guardarlo en su “array” correspondiente, pero surge un problema con esta situación.

No todas las aplicaciones tienen correo electrónico, algunas tienen ese campo vacío por lo que puede plantear un problema. En este caso, se ha creado un código para sustituir en el “array” de los correos un espacio en blanco en aquellos que no tengan, de no ser así, todos los “arrays” tendrían una longitud igual exceptuando el de los correos, que tendrían tantos menos como aplicaciones sin correo electrónico haya.

```

doc = message.doc[0]
for c in doc.child:
    shareUrl = c.shareUrl
    title = c.title
    url = shareUrl
    req = requests.get(url)
    statusCode = req.status_code
    aux = []
    apps.append(c.title)
    descargas.append(c.details.appDetails.numDownloads)
    actualizaciones.append(c.details.appDetails.uploadDate)
    ranking.append(c.aggregateRating.starRating)
    enlaces.append(c.shareUrl)
    if statusCode == 200:
        html = BeautifulSoup(req.text)
        correo = html.find_all('a',{'class':'dev-link'})
        for j,correo in enumerate(correo):
            href = correo.get('href')
            if not href.find("mailto"):
                mails.append(href)
        if len(apps) > len(mails):
            mails.append("no tiene")
    buscar = "mailto:"
    reemplazar = ""
    for t in range(len(mails)):
        mails[t] = mails[t].replace(buscar, reemplazar)

```

Ilustración 18. Código de extracción de datos (4): Obtención de información de la aplicación (4.2).

Una vez creados los “arrays” vacíos, se deben rellenar con la información mediante el siguiente bucle “for”. Previamente, se ha cogido la búsqueda realizada y se ha guardado en la variable “doc”. Esta variable es una lista que cuenta con las 250 aplicaciones que hay en la web de Google Play Store, por lo tanto, con la instrucción “for c in doc.child” se recorre la lista.

De cada aplicación se obtiene la URL, el título en las variables “shareUrl” y “title” correspondientes, además de obtener por segunda vez la URL en la variable “url”. El motivo de obtener dos veces la URL es porque una se usará para almacenar la información en la lista de enlaces, mientras que la otra se usará para poder obtener acceso al código HTML de la aplicación concreta, a través de scraping.

```

doc = message.doc[0]
for c in doc.child:
    shareUrl = c.shareUrl
    title = c.title
    url = shareUrl
    req = requests.get(url)

```

Ilustración 19. Código de extracción de datos (4): Obtención de información de la aplicación (4.3).

Con la función “get()” del módulo “requests” se realiza una petición HTTP a la página web para obtener su código HTML. Además del HTML, la función también aporta el “Status Code” de la página web. El “Status Code” indica si la página está operativa o si está fuera de servicio a través de códigos de 3 cifras, por ejemplo, el código “200” significa que está funcionando correctamente; el código “404” significa que no encuentra una página con esa URL; y el código “500” significa que ha habido un error interno del servidor. Se obtiene mediante la siguiente instrucción:

```

statusCode = req.status_code

```

Ilustración 20. Código de extracción de datos (4). Obtención de información de la aplicación (4.4).

Con el “Status Code” y el HTML de la página ya se puede scrapear la web. Antes de empezar el scraping, se almacenará los datos de la aplicación que ya se tienen gracias a GooglePlayAPI, como son: el título, el número de descargas, las actualizaciones, el número de estrellas y los enlaces.

```

apps.append(c.title)
descargas.append(c.details.appDetails.numDownloads)
actualizaciones.append(c.details.appDetails.uploadDate)
ranking.append(c.aggregateRating.starRating)
enlaces.append(c.shareUrl)

```

Ilustración 21. Código de extracción de datos (4). Obtención de información de la aplicación (4.5).

A través de la variable “c”, que es el índice que se le ha añadido al bucle para recorrer la lista de aplicaciones, se recoge los campos que interesan. En este caso, se puede decir que en la variable

“c” se guarda una aplicación cada vez, es decir, por cada pasada del bucle se tiene una aplicación distinta.

Ahora solo falta obtener el campo de “Correo electrónico”, para ello lo primero es comprobar que la página web responde: si el “Status Code” es igual a “200”, entonces se continúa con el código. En cualquier otro caso, el código falla y se tendría que volver a ejecutar el programa.

¿Qué motivos puede haber para que falle la conexión con la tienda de Google Play? Principalmente puede ser:

- La página web está en mantenimiento y durante cierto tiempo no está disponible.
- Problemas con la conexión de Internet, por ejemplo, se ha ido el Internet y no puedes navegar en ningún sitio hasta que vuelva.
- Otro motivo es que se haya escrito mal la URL, pero en este caso, como la obtenemos a través de una variable de Google, es poco común que falle por esta vía.

Comprobado que se conecta con la página web, el siguiente paso es crear un objeto de la clase “BeautifulSoup”, pasándole el HTML de la web.

```
if statusCode == 200:  
    html = BeautifulSoup(req.text)
```

Ilustración 22. Código de extracción de datos (4). Obtención de información de la aplicación (4.6).

Con ese objeto se va a poder adquirir, con los métodos de “BeautifulSoup”, los elementos necesarios de la web.

```
correo = html.find_all('a',{'class':'dev-link'})
```

Ilustración 23. Código de extracción de datos (4). Obtención de información de la aplicación (4.7).

La función “find_all” busca, en el código proporcionado previamente en la variable “html”, una clase determinada dentro de la etiqueta que se le pase, en este caso <a>. Por lo tanto se está buscando los enlaces cuya clase sea “dev-link”. Por último, se guarda la información en la variable “correo”, pero ¿qué hay realmente en dicha variable almacenado?

Dentro de la variable “correo” hay una lista de 3 elementos (enlaces), de forma que se pueda acceder a ellas a través de un bucle “for”.

```
<a class="dev-link"
  href="https://www.google.com/url?q=http://www.zaragoza.es/ciudad/viapublica/movilidad/c
  rel="nofollow" target="_blank">
  Visitar sitio web
</a>
<a class="dev-link" href="mailto:internet@zaragoza.es"
  rel="nofollow" target="_blank">
  Correo electrónico: internet@zaragoza.es
</a>
<a class="dev-link"
  href="https://www.google.com/url?q=https://www.zaragoza.es/sede/portal/politica-privaci
  rel="nofollow" target="_blank">
  Política de Privacidad
</a>
```

Ilustración 24. Información adicional: Datos dentro de la variable “correo”.

Ahora falta detectar cuál de los 3 enlaces tiene el correo electrónico de la aplicación, para ello se hará uso de otro bucle “for” que recorrerá la lista de correos. Gracias a la librería “BeautifulSoup”, proporciona una función que detecta el campo “href” de cada enlace.

Para recordar, anteriormente se dijo que para distinguir el correo electrónico de entre los 3 enlaces, se debía encontrar una palabra clave que lo identifique dentro del campo “href”, y se escogió la palabra “mailto”.

Dentro del bucle, una vez obtenidos cada enlace por separado, con la función de Python “find()” se identifica el enlace con la palabra “mailto” y será ese, únicamente, el que se almacene en el “array” de correos.

```
for j, correo in enumerate(correo):
    href = correo.get('href')
    if not href.find("mailto"):
        mails.append(href)
```

Ilustración 25. Código de extracción de datos (4). Obtención de información de la aplicación (4.8).

3.4.1. PROBLEMAS CON LAS APLICACIONES SIN CORREO

Llegado el momento de almacenar los correos, surge un problema bastante grave que condiciona el resultado de los datos. Existen aplicaciones que no tienen correo, o no se muestra en la información de la aplicación de la tienda Google Play, por lo tanto, se tienen “arrays” cuya longitud es menor y causará un problema a la hora de compararlos posteriormente.

Al final, se encuentra la solución de almacenar el texto “no tiene”, en el “array” de correos, en las aplicaciones sin correo, para eso, después de terminar el bucle “for” donde se detecta el enlace que posee o debería tener el correo, se escribe la siguiente instrucción:

```
if len(apps) > len(mails):  
    mails.append("no tiene")
```

Ilustración 26. Código de extracción de datos (4). Obtención de información de la aplicación (4.9).

Con esta instrucción se comprueba si la longitud del “array” donde se almacenan los títulos (“apps”) es mayor al del “array” donde se almacenan los correos (“mails”) (se daría el caso si aparece una aplicación donde no hay correo), en ese momento, se añade al final del “array” de correos el texto “no tiene” (¿Por qué al final del “array”? Porque es la posición en la que se encuentra el índice, es decir la última aplicación que hemos analizado, y la función “append()” añade al final del “array” lo que viene ideal para este caso).

3.4.2. ARREGLAR LOS CORREOS ADECUADAMENTE

Al obtener los correos con la instrucción “mails.append(href)” almacena todo el valor que tiene “href”, es decir, incluye la palabra clave “mailto”, por ejemplo:

- href="mailto:internet@zaragoza.es"

No es un problema exactamente, pero si se quiere obtener la información correctamente se deberá quitar del correo la palabra “mailto” o se almacenará así en la hoja de Excel. Para ello, usando la función “replace()” de Python, se reemplazará la palabra “mailto” por nada, es decir, como si se añadiera un espacio en blanco que no ocupa sitio.

```
buscar = "mailto:"
reemplazar = ""
for t in range(len(mails)):
    mails[t] = mails[t].replace(buscar, reemplazar)
```

Ilustración 27. Código de extracción de datos (4). Obtención de información de la aplicación (4.10).

En primer lugar, se guarda en la variable “buscar” la palabra que se quiere sustituir (“mailto”). En segundo lugar, se guarda en la variable “reemplazar” la palabra o el texto por el que se quiere reemplazar. Por último, se crea un bucle “for” que recorre el “array” de correos y, con la función “replace(buscar, reemplazar)” se sustituye el primer elemento de la función por el segundo, en cada aplicación del “array”, y se vuelve a guardar la información en el mismo “array” de correos (de forma que se sobrescriba, sin tener que crear otra variable nueva).

3.5. EXPORTACIÓN EN EXCEL

Para finalizar la parte del código, se debe exportar la información obtenida en una hoja Excel, para poder trabajar a partir de ella. Con la librería “xlwt” se podrá realizar de forma muy sencilla:

```
for s in range(len(apps)):
    (ws.row(s+1)).write(0, apps[s])
    (ws.row(s+1)).write(1, mails[s])
    (ws.row(s+1)).write(2, descargas[s])
    (ws.row(s+1)).write(3, actualizaciones[s])
    (ws.row(s+1)).write(4, ranking[s])
    (ws.row(s+1)).write(5, enlaces[s])

wb.save('Correos.xls')
```

Ilustración 28. Exportación en Excel de la información.

Mediante un bucle se recorre uno de los 6 “arrays” (¿Por qué solo uno? Porque todos tienen la misma longitud, por lo tanto, da igual cual se elija). En cada columna se va añadiendo cada valor de cada “array” completando fila por fila la hoja de Excel mediante la función “ws.row(s+1).write()”.

Como se puede observar, la función empieza en la segunda fila (“s+1”). Se debe a que en la primera fila están los títulos de cada columna.

Para terminar, se guarda la hoja de Excel mediante la función “wb.save()” donde se le añade un nombre que lo identifique.

4. FILTRADO DE APLICACIONES

Una vez se tenga la hoja del Excel rellena, hay que proceder a la fase de filtrado de aplicaciones. Esto se debe a que la mayoría de aplicaciones son de ayuntamiento, pero siempre hay alguna que no corresponde, a menudo suelen ser videojuegos. Hay que destacar que esta tarea ya se hace manualmente, por lo que hay que repasar las 250 aplicaciones que tiene el Excel.

Básicamente, lo que se tiene que fijar uno en esta tarea es en:

- **El título de la aplicación:** Es lo más identificativo posible en esta tarea, ya que la mayoría de aplicaciones de ayuntamiento tiene la palabra “ayuntamiento en sus nombres”, aunque suele haber excepciones, pero con esta opción se ahorra mucho tiempo.
- **El número de descargas:** Suele ser otro dato significativo. Las aplicaciones de ayuntamiento por lo general no superan las 10.000 descargas, por lo que, si alguna aplicación tiene más de 500.000 descargas, es posible que sea un videojuego.
- **El correo electrónico:** Este campo es menos fiable. La mayoría de aplicaciones de ayuntamiento tienen el correo del propio ayuntamiento como dirección electrónica, sin embargo, los videojuegos poseen correos electrónicos de las empresas que las crean. No obstante, muchas aplicaciones de ayuntamiento, por no decir todas, las han creado empresas externas, y por tanto contienen el correo de la empresa. El tema es que varios ayuntamientos no cambian ese correo electrónico y, por lo tanto, son difíciles de distinguir.

Algunos ejemplos de aplicaciones que se han descartado son:

NOMBRE DE LA APLICACION	CORREO ELECTRONICO	NUMERO DE DESCARGAS
Trucos Clash of Clans: COC Free	updraftlabs@gmail.com	500.000+
Los Del Rio music	info@wecromapp.com	500+
Castle Clash: Era de Bestias	Help.castle.es@igg.com	5.000.000+
Nueva actualización COC base	ngabase.id@gmail.com	1.000+
Foster's Hollywood	franciscogr@wanthumb.es	1.000.000+
coc base de TH 4-9	auravelin2@gmail.com	100.000+
Mapas base COC TH 9	zulkifiamin142@gmail.com	10.000+
Clash of Lords 2: Español	help.lords.es@igg.com	1.000.000+
Walking Dead: Road to Survival	twdsupport@scopely.com	10.000.000+
Clash of Clans	gp-info@supercell.com	100.000.000+
Bases para Clash of Clans	andrew.corpa@gmail.com	100.000+
Gladiator Heroes	gladiatorheroes@generagames.	50.000+

Ilustración 29. Aplicaciones descartadas.

Además de las aplicaciones de videojuegos, hay muchas otras que no son realmente las que se buscan para este proyecto, por ejemplo, aplicaciones de aparcamientos, mapas de ciudades 3D, aplicaciones de carriles bici, de museos y monumentos, etc.

Las aplicaciones que se van a tratar en este proyecto son aquellas que tienen las funciones propias de un ayuntamiento, algunas de ellas son:

- Restaurantes y hoteles.
- Turismo.
- Información general.
- Indicaciones de cómo llegar.
- Servicios municipales.

Un poco en general, aunque más adelante se verán muchas más funciones que los ayuntamientos suelen tener. A simple vista no se puede distinguir perfectamente si son aplicaciones de ayuntamientos o no, salvo las muy claras, mencionadas anteriormente. Como no hace falta investigar todas, se destacarán todas aquellas que no tengan la palabra clave ayuntamiento en su título. De esa forma, es seguro que todas las aplicaciones que se analicen a partir de ahora tendrán que ser relacionadas con ayuntamientos.

Algunos ejemplos de aplicaciones de ayuntamiento son:

NOMBRE DE LA APLICACION	CORREO ELECTRONICO	NUMERO DE DESCARGAS	ULTIMA ACTUALIZACION	NUM DE ESTRELLAS
Ayuntamiento de Albox	rafaelterantorralbo@gmail.c	100+	16 ago. 2016	4,199999809
Ayuntamiento de Añover de Taj	soporte@inbox-mobile.com	100+	11 feb. 2016	4,400000095
Ayuntamiento de Caniles	info@vip-apps.es	100+	25 ene. 2016	5
Ayuntamiento de Tías	info@social-val.com	100+	4 ene. 2016	4,857142925
Ayuntamiento de Baza	baza.appmipueblo@gmail.c	100+	6 feb. 2015	4,818181992
010 Ayuntamiento Valladolid	info@todovalladolid.com	100+	3 ago. 2014	1,799999952
Ayuntamiento de Los Gallardos	rafaelterantorralbo@gmail.c	100+	19 jul. 2016	5
Ayuntamiento de Porcuna	soporte@vip-apps.es	100+	25 ene. 2016	4,666666508
Ayuntamiento de Moclín	moclin.app@gmail.com	100+	24 dic. 2015	4,333333492
Ayuntamiento de San Bartolom	info@social-val.com	100+	15 nov. 2016	4,454545498
Ayuntamiento de Boecillo	ayuntamiento@boecillo.es	100+	13 dic. 2016	4,857142925
Ayuntamiento de Las Gabias	info@aytolasgabias.org	100+	1 jul. 2015	2,571428537
Ayuntamiento de Dílar	dilar.app@gmail.com	100+	12 dic. 2016	4,636363506
Ayuntamiento Espartinas	dev.bluumi.sa@gmail.com	100+	7 may. 2013	2,900000095
Ayuntamiento de Cuntis	support@evelb.es	100+	12 jun. 2015	3,176470518
Ayuntamiento Guadalupe Inform	info@bandomovil.com	100+	22 nov. 2016	4,625
Ayuntamiento de Coaña	animacion.ayuntamientode	100+	2 jun. 2013	5
Ayuntamiento de Valleseco	eric@reskyt.com	100+	4 nov. 2015	4
Ayuntamiento de Ribaforada	eric@reskyt.com	100+	11 nov. 2015	3,666666746
Ayuntamiento Colindres	google@estelacantabra.co	100+	10 may. 2016	5
Ayuntamiento de Fondón	soporte@inbox-mobile.com	100+	10 ago. 2016	5
Ayuntamiento de Algarinejo	algarinejo.app@gmail.com	100+	13 dic. 2016	4,538461685
Ayuntamiento	master.26@mail.ru	100+	5 dic. 2016	4,800000191
Ayuntamiento de Loja	info@appbarata.es	100+	3 oct. 2014	4,125
Ayuntamiento de Valmojado	soporte@inbox-mobile.com	100+	20 abr. 2016	5
Ayuntamiento de Cebreros	victor.sanchez.yonte@ovny	100+	13 oct. 2016	4,900000095
Ayuntamiento de Cabrerizos	eric@reskyt.com	100+	11 nov. 2015	4,833333492
Ayuntamiento de Tabernas	tabernas.app@gmail.com	100+	30 sept. 2016	4,857142925
Ayuntamiento de Guillena	dev.bluumi.sa@gmail.com	100+	16 may. 2014	4
Ayuntamiento de Doñinos	eric@reskyt.com	100+	29 ene. 2016	4
Ayuntamiento de Monachil	monachil.app@gmail.com	100+	8 abr. 2015	4,888888836
Ayuntamiento de Trabuco	villanuevadeltrabuco.app@c	100+	30 ago. 2014	3,400000095
Ayuntamiento de LEÓN	contact@sikiwis.com	100+	1 feb. 2015	0

Ilustración 30. Aplicaciones de ayuntamiento válidas.

Una vez se tenga el Excel actualizado con las aplicaciones de ayuntamiento válidas, mediante los filtros aplicados, el siguiente paso será analizar algunas aplicaciones e identificar posibles funciones.

5. ANÁLISIS E IDENTIFICACIÓN DE FUNCIONALIDADES

En esta sección se explicará el proceso que se ha seguido para el análisis de las funcionalidades y se verán tres ejemplos de aplicaciones para visualizar la diferencia que hay entre ellas. Esta tarea sigue siendo manual, por lo que el proceso consistirá en descargar la aplicación móvil e investigar, navegando por todas las funciones que tiene. Estas funciones se recopilarán en un Excel que servirá como plantilla para identificar y planificar las funcionalidades que serán útiles para una nueva aplicación de ayuntamiento.

Para elegir las aplicaciones a analizar, se ha pensado que la mejor forma de tener un conocimiento global será, seleccionar una que tenga muchas descargas y otra con pocas descargas; una con muchas estrellas y otra con pocas estrellas; y por supuesto, alguna ciudad importante Valencia.

Aunque se han analizado alrededor de 10 aplicaciones en total, en este documento se mostrarán únicamente tres aplicaciones. En realidad, lo que importa en este apartado es identificar las funciones de las aplicaciones, aunque se crea que no tengan un uso importante, e introducir las en el Excel.

Una vez recopiladas, ya será momento de agruparlas por categorías y, si se da el caso, eliminar algunas funciones.

5.1. APLICACIÓN DEL AYUNTAMIENTO DE CARTAGENA

La aplicación de Cartagena se ha escogido por tener un número de estrellas muy alto, de 4.3/5. Es la primera aplicación que se analiza por lo que en el Excel aparecerán todas las funciones. A partir de la segunda, se podrá ver como la hoja de Excel toma forma y se visualizan las diferencias entre una y otra.



Ilustración 31. Aplicación de Cartagena: Menú de inicio.

A primera vista, tiene 3 funcionalidades generales operativas: Actualidad, Turismo e Información general.

Dentro de Actualidad, tiene 3 opciones: Noticias, Agenda y Tablón.



Ilustración 32. Aplicación de Cartagena: Menú de actualidad

En el menú de Turismo, se encuentran ya opciones más interesantes relacionadas con los ayuntamientos:



Ilustración 33. Aplicación de Cartagena: Menú de turismo.

Por último, en el menú de Información general:



Ilustración 34. Aplicación de Cartagena: Menú de información general.

A primera vista es una aplicación bastante completa, con todas las funciones que deberían aparecer en una aplicación de ayuntamiento, pero aún es pronto para saber si es correcta o no. Dependerá del análisis de las otras aplicaciones.

Un problema que se ha encontrado en esta aplicación ha sido la actualización de información. En una aplicación de este tipo es muy importante que las noticias se vayan actualizando, o descartar el apartado de noticias.

5.2. APLICACIÓN DEL AYUNTAMIENTO DE MIERES

La aplicación del ayuntamiento de Mieres se ha escogido por tener un número muy pequeño de descargas, un poco más de 500, según el Excel. Aunque parece que por sus pocas descargas no esté a la altura de las otras aplicaciones, a continuación, se verá que todo lo contrario. Es muy

completa, con muchas funciones y bien organizadas, de hecho, es de las mejores aplicaciones que se ha analizado en este proyecto.



Ilustración 35. Aplicación de Mieres: Menú de inicio.

El menú de inicio consta de 4 funciones principales: Tablón de Anuncios, noticias, Agenda y “Más”. Las imágenes están incluidas en un “slider” que va rotando cada 5 segundos (de muy alta calidad la resolución).

En esta aplicación, las noticias sí que están actualizadas, así como el tablón de anuncios y la agenda. En cuanto a la opción de “Más”, se abre en otro menú con varias funciones extras más propias de un ayuntamiento:

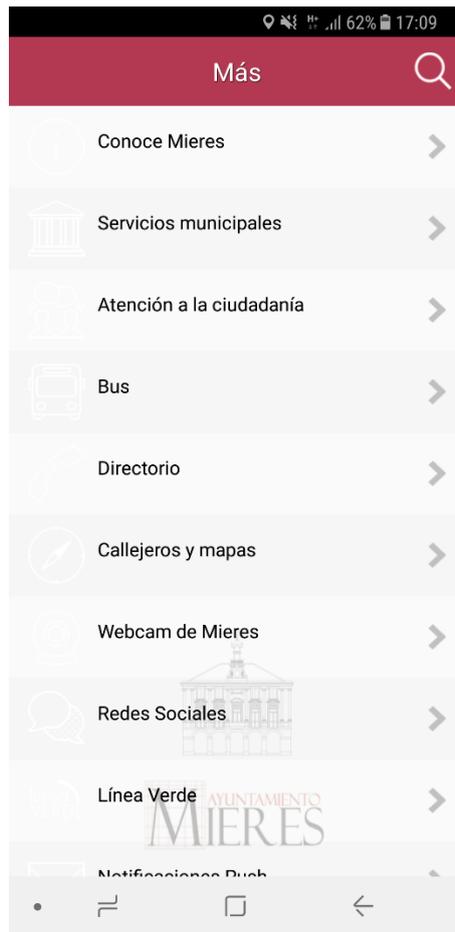


Ilustración 36. Aplicación de Mieres: Menú de "Más" (1).

Se puede ver como aparecen apartados un poco ambiguos como “Conoce Mieres” o “Servicios municipales” que a primera impresión no se sabe lo que te puede ofrecer, sin embargo, se despliegan en otros menús con una visualización mucho más clara.

También se encuentra un detalle muy importante: “Atención a la ciudadanía”, que indica información personalizada sobre trámites y gestiones municipales, recepción de consultas y sugerencias, petición de citas a concejales o comunicación de incidencias.

Cuenta con un horario de autobuses, directorio telefónico y un callejero (imprescindible en un ayuntamiento), además de otras funciones de interés como webcam, línea verde o la posibilidad de activar notificaciones en tu móvil.

Un apartado que hay que destacar es el de las redes sociales, muy común en el día a día de muchas personas. Poseen las direcciones de Facebook/Twitter de varios establecimientos, no solo del ayuntamiento sino también, de bibliotecas, oficinas, casas de encuentros, etc.

5.2.1. FUNCIONES ESENCIALES DE UN AYUNTAMIENTO

Como se ha mencionado antes, existen dos apartados un poco ambiguos pero que se abrían en otros dos menús, que se van a analizar en este apartado en concreto ya que, poseen las funciones esenciales que todo ayuntamiento debe tener, y resulta de interés recalcarlas para el proyecto.

El primer menú a analizar es el de “Conoce Mieres”, que se divide en:



Ilustración 37. Aplicación de Mieres: Menú de "Más" (2) - Conoce Mieres

Se puede ver como destaca la visualización en imágenes. Consta de funciones típicas de un ayuntamiento:

- **Museos y centro de interpretación:** Se pueden encontrar varios sitios donde hacen exposiciones/interpretaciones.
- **Oficina de Turismo:** El lugar donde informarse. Cuenta con la dirección de la oficina, teléfono y correo electrónico de contacto, y los horarios de apertura.
- **Mapas y callejeros:** Es el típico mapa donde puedes navegar por las calles.
- **Dónde comer:** Es un menú donde aparece un listado de restaurantes divididos en distintos tipos como asadores, bares o hamburgueserías, entre otras. Y dentro de cada sección, el sitio con su teléfono de contacto.
- **Dónde dormir:** Otro menú con los distintos hoteles donde se puede reservar.
- **Patrimonio y Naturaleza:** Indica, por encima, el patrimonio artístico, etnográfico e industrial de Mieres, así como paisajes protegidos y rutas de senderismo.
- **Geografía e Historia:** Un apartado donde conocer la historia de Mieres y la situación geográfica del municipio.
- **Nuestros pueblos:** Es un listado de pueblos que pertenecen al municipio de Mieres.
- **Galerías fotográficas:** Este apartado no está operativo.

El segundo menú a analizar es el de “Servicios municipales”, que se divide en:

- **Bienestar Social:** Información sobre el centro de servicios sociales, residencias, centro de apoyo, etc.
- **Consumo y comercio:** Información sobre la oficina de información al consumidor.
- **Cultura:** Listado de bibliotecas de Mieres, así como casas de cultura y centros de estudios.
- **Deportes:** Instalaciones deportivas de Mieres.
- **Educación:** Escuelas y colegios, así como la universidad Campus de Mieres.
- **Empleo:** Información sobre programas de empleo y selección de ofertas de empleo público y privado.
- **Festejos:** Información sobre fiestas que se celebran en Mieres.
- **Juventud e Infancia:** Datos sobre la oficina de información juvenil, donde se puede encontrar información sobre empleos, subvenciones, cursos, etc.
- **Medio Rural:** Oficina de medio rural.

- **Medio Ambiente:** Oficina de Medioambiente.
- **Mujere Igualdad:** Información de centros que asesoran a la mujer, así como la agencia de igualdad de oportunidades.
- **Salud:** Listado de centros médicos y hospitales de Mieres.
- **Seguridad Ciudadana:** Información sobre oficinas de seguridad como: policía local, bomberos, protección civil, etc.



Ilustración 38. Aplicación de Mieres: Menú de "Más" (3) - Servicios municipales

Como se ha mencionado anteriormente, parecía una aplicación bastante pobre, debido al número de descargas que había recibido, pero después de analizarla, se puede afirmar que es una aplicación bastante completa y seguro que se tomarán muchas funciones de ella y servirá como ejemplo para otras aplicaciones.

5.3. APLICACIÓN DEL AYUNTAMIENTO DE VALENCIA

Por último, se analizará una aplicación con bastantes descargas (más de 10.000), un rating de 3,9/5 de puntuación y una ciudad muy conocida como es Valencia. Ha llegado el momento de ver si una aplicación de una ciudad con más fama (es decir, más conocida) posee mejores funciones y está mejor diseñada que las que se han visto anteriormente.

Esta aplicación tiene un menú bastante diferente al del resto que se han visto:

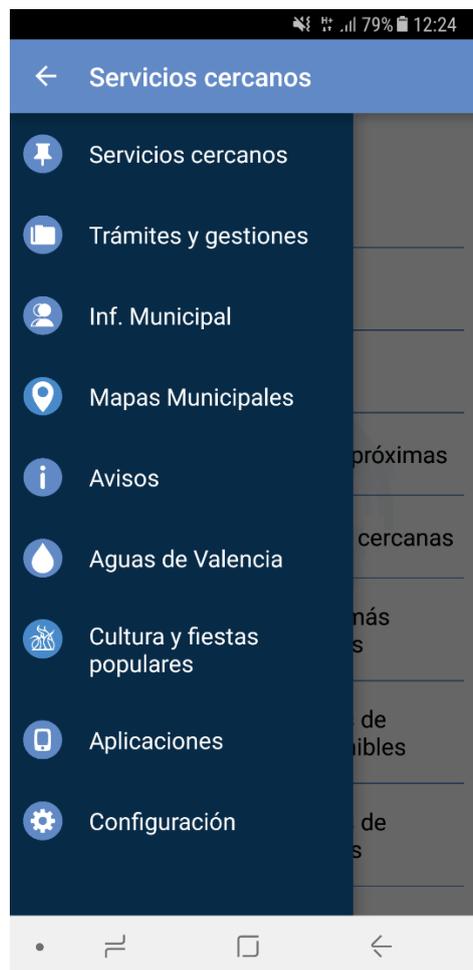


Ilustración 39. Aplicación de Valencia: Menú de funciones.

En él se puede observar que aparecen algunas de las funciones típicas de los ayuntamientos, pero, además, funciones extras como “Servicios cercanos”, “Trámites y gestiones” o “Aguas de Valencia”. Algunas opciones se verán como son subjetivas de la propia ciudad de Valencia, por

lo que no se podrán tomar en cuenta a la hora de incluirlas en el Excel, a no ser que se puedan englobar en algún grupo más general.

Aquí ya no aparecen las funciones de “Donde comer” ni “Donde dormir” típicas de las aplicaciones de ayuntamiento. Si es verdad que aparecen los mapas, culturas y fiestas, pero se queda corta de información a simple vista.

En el menú de “Servicios cercanos” encontramos apartados como:

- **Geoportal:** Redirecciona a la página web del geoportal de valencia.
- **Estado del tráfico:** Indica si hay calles cortadas.
- **¿Qué hay cerca de mí?:** Muestra un listado de lugares que puede buscar cerca de tu posición, desde 300m hasta 5km de distancia. La variedad se divide entre restaurantes, instalaciones sanitarias, bibliotecas, policía, transporte, fallas, etc.

Hay más apartados del tipo “establecimiento más cercano” que se han resumido en el punto de “¿Qué hay cerca de mí?” pero se entiende que se han sacado fuera para tenerlas más a mano.

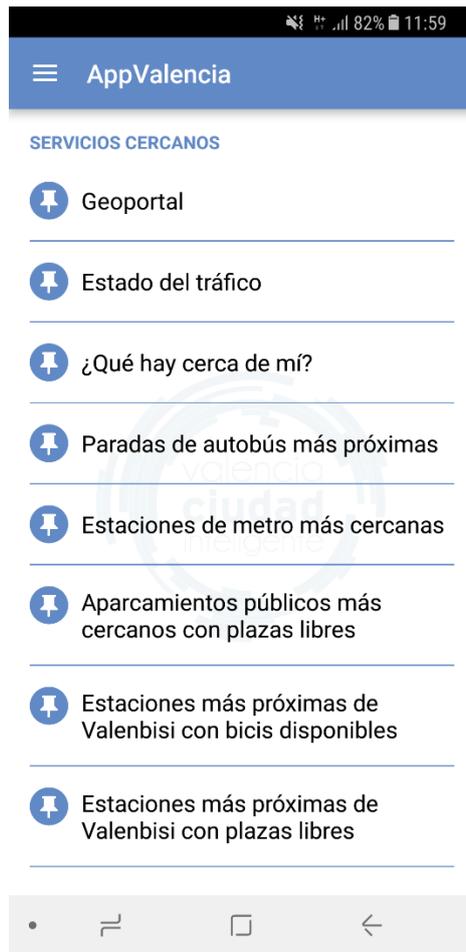


Ilustración 40. Aplicación de Valencia: Menú de servicios cercanos.

La sección de “Trámites y gestiones” sirve, más que nada, para consultar documentos administrativos y pedir cita en las oficinas correspondientes.



Ilustración 41. Aplicación de Valencia: Menú de trámites y gestiones.

El apartado de “Información Municipal” muestra las últimas noticias de diversos temas de la ciudad, la agenda de eventos, y varios documentos de campañas municipales.

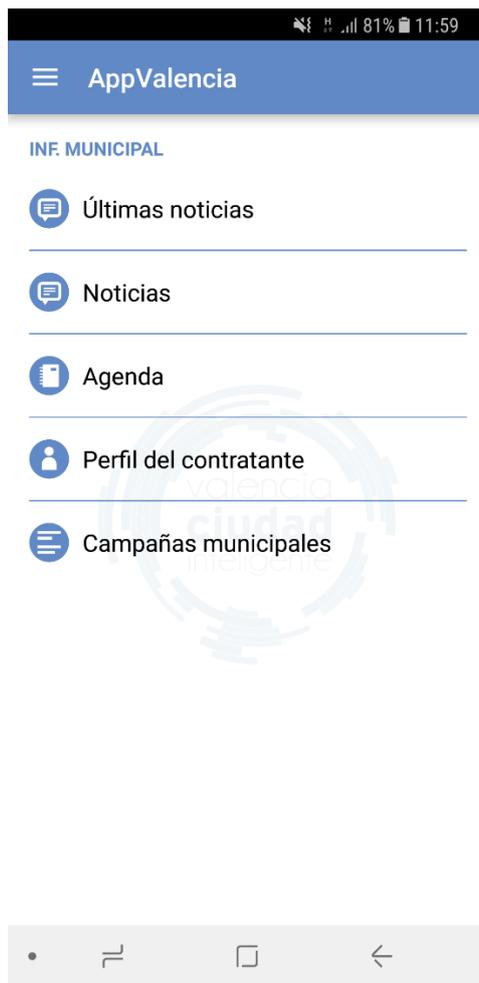


Ilustración 42. Aplicación de Valencia: Información municipal.

En cuanto a la opción de “Aguas de Valencia”, este apartado ofrece suscribirse a EMIVASA (la empresa de aguas de valencia) para notificar posibles fugas de agua, cortes de suministro, así como el importe y consumo del último periodo de facturación.

El apartado de avisos, no es la típica opción de notificaciones que se han visto anteriormente, sino una lista de avisos recientes sobre diversos temas de importancia, como tráfico o salud. Son como unas noticias de última hora que pueden resultar de gran ayuda si uno va a salir a la calle o coger el coche.



Ilustración 43. Aplicación de Valencia: Listado de avisos.

El apartado de “Cultura y fiestas populares” refleja la información de las fiestas como se han visto en las anteriores aplicaciones. Una variedad de la aplicación de Valencia son las Fallas. Como patrimonio de la humanidad le han dedicado bastante espacio dentro de la aplicación, con una agenda de las fallas, castillos de fuego, ofrendas, fallas cercanas, etc.



Ilustración 44. Aplicación de Valencia: Menú de cultura y fiestas populares.

Por último, está el apartado de “Aplicaciones”, una opción que no habíamos visto en ninguna otra aplicación analizada con anterioridad. En esta sección, se encuentran aplicaciones móviles que se pueden descargar y son de interés público. Hay de diferentes tipos: de museos, transporte, cocina, turismo, fallas, etc.

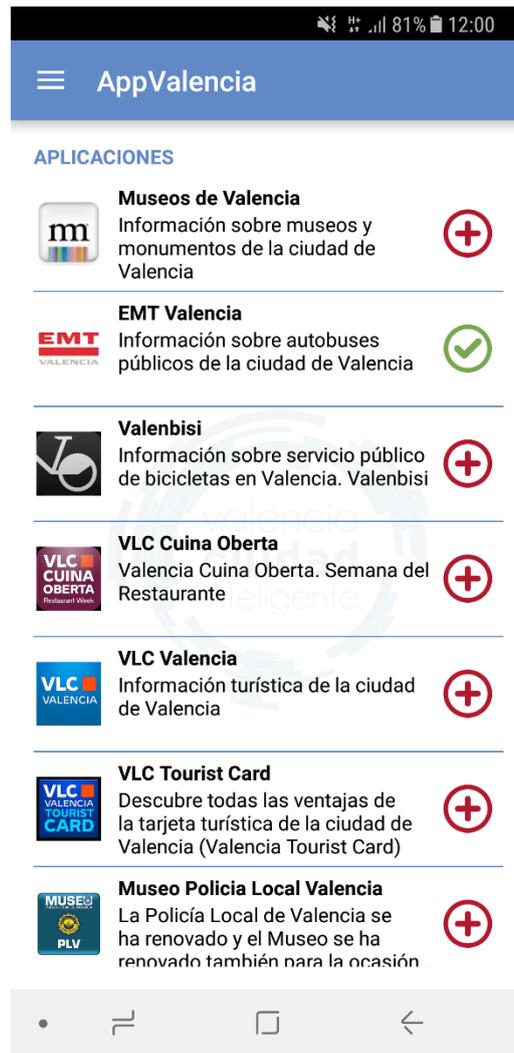


Ilustración 45. Aplicación de Valencia: Menú de aplicaciones.

De la aplicación de Valencia se puede destacar dos apartados novedosos en cuanto a diseño se refiere: “¿Qué hay cerca de mí?” Y “Aplicaciones”. A pesar de ser novedosas las funciones, únicamente se puede añadir al Excel la de descargar aplicaciones de otro tipo que pueden resultar de interés, ya que la opción de “¿Qué hay cerca de mí?”, aunque su diseño fuera bastante moderno y útil, las funciones que buscaba a partir de la posición ya se han introducido en el Excel.

Aun así se ha sacado en claro que, la aplicación del ayuntamiento de Valencia cuenta con un buen diseño, buenas funcionalidades, y apartados de gran utilidad para sus habitantes.

6. SELECCIÓN FINAL DE FUNCIONES

Una vez recopilada toda la información en una hoja de Excel, se han rediseñado algunas funciones que se repetían con distinto nombre o que se asemejaban bastante y se ha optado por juntar para ahorrar espacio en la pantalla. Hay que tener en cuenta que la pantalla del móvil es reducida, por lo que se tiene que optimizar al máximo el espacio sin que resulte demasiado cargante (aunque ese proceso no se tratará en este proyecto, únicamente la selección de funciones que se han de añadir, o es conveniente, en una aplicación de ayuntamiento).

Después de organizar las funciones, quedan un total de 59:

Actualidad/Noticias	Configuración
Agenda	Servicios municipales
Turismo:	Cultura/Bibliotecas
Playas	Educación/Escuelas/Universidad
Dónde comer/Restaurantes	Fiestas locales/Ferías
Dónde dormir/Alojamientos	Información juvenil
Temporada	Bienestar social/servicios sociales
Actividades (deportivas)	Oficina de Medio Rural
Itinerarios turísticos	Oficina de Medio Ambiente
Gastronomía local	Mujer e Igualdad
Museos	Seguridad ciudadana/Policia/Bomberos
Información general/Destacados:	Redes sociales:
Incidencias	Facebook
Cursos de formación/Empleo	Twitter
Salud/Farmacias/Centros de salud	Notificaciones
Cita previa	Plenos
Teléfonos de interés/Directorio	Consulta ciudadana/Participación
Transporte/Bus/Metro/Taxi/Bici	Iniciar sesión/Registrarse
El tiempo	Enlace compartir
Cines/Bares/Discootecas/Parques	Valoración
Mercadillos/Consumo y Comercio	Corporación Municipal/Trabajadores
Aparcamientos	Objetos perdidos
Wifi/Puntos Wifi	Tablón de anuncios/Profesionales
Mapas/Cómo llegar	Radio/Televisión
Historia	Idioma
Geografía/Pueblos/Rutas	FAQ/Término y condiciones
Calendario laboral	Cerca de mí
Servicios online/página web	Aplicaciones
Actualizar	Opción Salir
	Audioguías

Ilustración 46. Funciones almacenadas en Excel.

Para analizar las funciones que se deben incluir en las aplicaciones, se va a utilizar una técnica simple de comparación, es decir, se compararán las funciones actualizadas en las diferentes aplicaciones. Esto servirá en primer lugar para indicar que funciones coinciden en la mayoría de aplicaciones y, en segundo lugar, para saber cuál es la aplicación que mejor representa nuestro modelo.

Para ellos se colocan las funciones en una columna y en las sucesivas las aplicaciones. Entonces se van marcando con un color si coincide que dicha función está contenida en la aplicación correspondiente. Saldrá un resultado como el siguiente:

Funciones/Servicios	Cartagena	Mieres	La Oliva	Benavente	Dos Torres	Marbella	Dumbria	Chinchón	Guadalajara	Valencia	Girona
Actualidad/Noticias											
Agenda											
Turismo:											
Playas											
Dónde comer/Restaurantes											
Dónde dormir/Alojamientos											
Temporada											
Actividades (deportivas)											
Itinerarios turísticos											
Gastronomía local											
Museos											
Información general/Destacados:											
Incidencias											
Cursos de formación/Empleo											
Salud/Farmacias/Centros de salud											
Cita previa											
Teléfonos de interés/Directorio											
Transporte/Bus/Metro/Taxi/Bici											
El tiempo											
Cines/Bares/Discotecas/Parques											
Mercadillos/Consumo y Comercio											
Aparcamientos											
Wifi/Puntos Wifi											
Mapas/Cómo llegar											
Historia											
Geografía/Pueblos/Rutas											
Calendario laboral											
Servicios online/página web											
Actualizar											
Configuración											

Ilustración 47. Comparación de las funciones en las diversas aplicaciones analizadas (1).

Funciones/Servicios	Cartagena	Mieres	La Oliva	Benavente	Dos Torres	Marbella	Dumbria	Chinchón	Guadalajara	Valencia	Girona
Servicios municipales											
Cultura/Bibliotecas											
Educación/Escuelas/Universidad											
Fiestas locales/Ferías											
Información juvenil											
Bienestar social/servicios sociales											
Oficina de Medio Rural											
Oficina de Medio Ambiente											
Mujer e Igualdad											
Seguridad ciudadana/Policia/Bomberos											
Redes sociales:											
Facebook											
Twitter											
Notificaciones											
Plenos											
Consulta ciudadana/Participación											
Iniciar sesión/Registrarse											
Enlace compartir											
Valoración											
Corporación Municipal/Trabajadores											
Objetos perdidos											
Tablón de anuncios/Profesionales											
Radio/Televisión											
Idioma											
FAQ/Término y condiciones											
Cerca de mí											
Aplicaciones											
Opción Salir											
Audioguías											

Ilustración 48. Comparación de las funciones en las diversas aplicaciones analizadas (2).

Tras los resultados obtenidos, se ve como las primeras funciones las tiene casi todas incluidas en la aplicación de Cartagena, ya que fue la primera en analizar, sin embargo, después se ve como la otra mitad de funciones no se incluyen.

Al principio del análisis es normal que esto suceda, la dos o tres primeras aplicaciones se usan para almacenar casi todas las aplicaciones que debe tener una aplicación de ayuntamiento, aunque si se ve alguna función que coincide se le va añadiendo. Poco a poco, según se va cambiando de aplicación, se van añadiendo funciones que van surgiendo.

Como se ha mencionado anteriormente, la elección de las aplicaciones ha sido variada, algunas se han cogido por el número de descargas, otra por el número de estrellas y otras por la importancia de la ciudad en la península.

Si bien, algunas aplicaciones tienen muchas funciones incluidas, hay otras que no poseen apenas. Esto puede deberse a diversas situaciones:

- La empresa contratada hizo un trabajo simple, sin hacer penas investigación en que funciones añadir, o hizo una copia de alguna otra aplicación mal y corriendo.

- Otra razón puede ser que la ciudad, pueblo o municipio sea muy pequeño y no cuente con tanta información que aportar, lo que se termina reflejando en el tamaño y contenido de la aplicación.
- Otro motivo son que algunas de las funciones sean de una localización determinada que la situación geográfica de la ciudad, pueblo o municipio impida tener ese beneficio, por ejemplo, playa, centro médico, escuelas, cine, etc.

Una vez terminada la comparación, se ha creado una columna extra para indicar, con un color diferente, que funciones entrarían dentro del modelo planteado. En resumen, de las 59 funciones que se han obtenido, se han quedado en 38.

Actualidad/Noticias	
Agenda	
Turismo:	
Playas	
Dónde comer/Restaurantes	
Dónde dormir/Alojamientos	
Temporada	
Actividades (deportivas)	
Itinerarios turísticos	
Gastronomía local	
Museos	
Información general/Destacados:	
Incidencias	
Cursos de formación/Empleo	
Salud/Farmacias/Centros de salud	
Cita previa	
Teléfonos de interés/Directorio	
Transporte/Bus/Metro/Taxi/Bici	
El tiempo	
Cines/Bares/Discootecas/Parques	
Mercadillos/Consumo y Comercio	
Aparcamientos	
Wifi/Puntos Wifi	
Mapas/Cómo llegar	
Historia	
Geografía/Pueblos/Rutas	
Calendario laboral	
Servicios online/página web	
Actualizar	
Configuración	

Ilustración 49. Funciones escogidas para la plantilla (1).

Servicios municipales	
Cultura/Bibliotecas	
Educación/Escuelas/Universidad	
Fiestas locales/Ferias	
Información juvenil	
Bienestar social/servicios sociales	
Oficina de Medio Rural	
Oficina de Medio Ambiente	
Mujer e Igualdad	
Seguridad ciudadana/Policia/Bomberos	
Redes sociales:	
Facebook	
Twitter	
Notificaciones	
Plenos	
Consulta ciudadana/Participación	
Iniciar sesión/Registrarse	
Enlace compartir	
Valoración	
Corporación Municipal/Trabajadores	
Objetos perdidos	
Tablón de anuncios/Profesionales	
Radio/Televisión	
Idioma	
FAQ/Término y condiciones	
Cerca de mí	
Aplicaciones	
Opción Salir	
Audioguías	

Ilustración 50. Funciones escogidas para la plantilla (2).

7. RESULTADOS

Como se ha visto, el número de funciones se ha reducido considerablemente, pero, ¿por qué se han descartado algunas funciones y otras no? La explicación depende de cada una, en general, porque no se ha creído que esa función concordaba con el modelo de aplicación de ayuntamiento, al menos, en una aplicación estándar.

Por ejemplo, la función de “Playas” es muy subjetivo, habrá ciudades que tengan y otras no; esa está clara de porqué se ha descartado. Lo mismo ocurre con “Temporada”, habrá ciudades que no tengan ese evento de temporada, o que no tengan algo típico que se pueda ofrecer. Más de lo mismo con la “Gastronomía local”.

Habrá ciudades, pueblos o municipios pequeños que únicamente cuenten con una escuela en todo el territorio, por ese motivo, se ha descartado los “Cursos de formación/Empleo”.

Lo mismo pasa con la sanidad, puede ocurrir que el hospital no cuente con mucho presupuesto y no tengan programas modernos, por lo cual, se ha descartado la “Cita previa” del médico.

En cuanto a la historia, está muy bien que la ciudad, pueblo o municipio tenga su historia, pero ocupar espacio en la aplicación móvil para ello, cuando se podría colocar otras funciones más importantes es absurdo, por lo tanto, se ha descartado también la función de “Historia” (este apartado se ve más como dentro de una página web que de una aplicación móvil).

Habrá lugares que no tengan un calendario laboral propio, por lo que usen el calendario laboral nacional, por ese motivo también se ha descartado el “Calendario laboral”.

En cuanto a los “Servicios online/páginas web”, no todos contarán con página web y mucho menos con servicios online, ya que una es directamente proporcional a la otra, por lo tanto, se ha descartado esa función.

Ocurre lo mismo con “Cultura/Biblioteca”, “Información juvenil”, “Oficina de Medio Rural”, “Oficina de Medio Ambiente”, “Mujer e Igualdad”, “Plenos”, “Objetos perdidos”, “Radio/Televisión” y “Aplicaciones”. Estas funciones son más comunes en aplicaciones de ciudades donde vive mucha gente y hay variedad de personas, si se refiere a un pueblo pequeño, no ofrecerá ninguno de estos servicios.

En cuanto a las “Redes sociales”, si se observa bien, se han dejado, pero no se han incluido ni “Facebook” ni “Twitter”. ¿Qué se quiere decir con esto? Pues que incluir las redes sociales es algo importante en la actualidad, pero apostar por una u otra no es elección nuestra.

Por último, la opción de “Iniciar sesión/Registrarse” es algo que no se aconseja en ningún momento. Aunque en algunas aplicaciones sea conveniente, no hay que olvidar que se están analizando aplicaciones de ayuntamiento. Si cada vez que una persona quiere entrar en la aplicación tiene que iniciar sesión, al final se cansaría y dejaría de entrar en ella, por lo tanto, no se recomienda incluirla.

En cuanto al contenido del proyecto, concretamente, la información obtenida de las aplicaciones de ayuntamiento, según los datos analizados, las funciones que más se suelen utilizar son:

- Noticias.
- Agenda.
- Turismo.
- Dónde comer/Restaurantes.
- Dónde dormir/Alojamientos.
- Itinerarios turísticos.
- Incidencias.
- Transporte.
- Mapas/Cómo llegar.
- Fiestas locales/ferias.

Sin embargo, las funciones que menos se han utilizado han sido:

- Playas.
- Temporada.
- Gastronomía local.
- Información general/Destacados.
- Cursos de formación/Empleo.
- Cita previa (médico).
- Cines/Bares/Discootecas/Parques.
- Aparcamientos.
- Calendario laboral.
- Actualizar.

- Educación/Escuelas/Universidad.
- Bienestar social/servicios sociales.
- Oficina de Medio Rural.
- Oficina de Medio Ambiente.
- Mujer e Igualdad.
- Iniciar sesión/Registrarse.
- Enlace compartir.
- Valoración.
- Corporación Municipal/Trabajadores.
- Objetos perdidos.
- Tablón de anuncios/Profesionales.
- Radio/Televisión.
- FAQ/Términos y condiciones.
- Cerca de mí.
- Aplicaciones.
- Audioguías.

Para finalizar, hay que destacar que ninguna aplicación analizada consta de una opción de “Salir”. Parece algo lógico que se incluya, aunque podría ser por el diseño de aplicación móvil que se sale con la del botón “Atrás” del móvil, aun así, se ha comprobado con muchas otras aplicaciones móvil (no de ayuntamiento) que si tenían dicha opción.

8. MEJORAS PROPUESTAS POR DESARROLLAR

Además de comparar las aplicaciones de ayuntamientos, se ha querido profundizar más en el proyecto e incluir más finalidades, para que sirva como posible negocio. Se ha pensado en guardar el correo electrónico de las aplicaciones, como se vio en la parte de extracción de datos, y en aquellas aplicaciones que no concuerden con el modelo planteado, porque falten funciones importantes que cualquier ayuntamiento tendría, mandarles un correo proponiéndoles la oferta de arreglar la estructura y el diseño de la aplicación.

Esto plantea un problema, y es que no todos los correos almacenados eran de los ayuntamientos, hay algunas aplicaciones que mantiene el correo electrónico de contacto de las empresas que las crearon, por lo que resulta más complicado contactar con esos ayuntamientos en concreto, aun así, buscando por Internet se encontraría fácilmente una dirección electrónica o un teléfono de contacto, aunque de la otra forma sería mucho más rápido.

En el caso de tener todos los correos almacenados, se podría hacer un programa en Python que leyera la hoja de Excel y cogiera el campo de "Correo". Entonces se le mandaría un mensaje estándar a cada ayuntamiento por si le interesa mejorar su aplicación de móvil, como una especie de spam.

Otra posible mejora sería ofrecer un servicio Premium con aquellas funciones que se han descartado o incluir nuevas para cada territorio en concreto. De esa forma, no se de aprovecha toda la información obtenida, sino que se reorganiza para sacarle un beneficio económico.

9. CONCLUSIONES

En la actualidad, el móvil, junto con el ordenador y la Tablet, son los dispositivos tecnológicos más utilizados. Aun así, el móvil ofrece un servicio que ni el ordenador o la Tablet ofrecen, y es el conectarse a Internet desde un dispositivo pequeño y portable, lo que le ha hecho destacar y que, cada vez más gente se una a esta moda.

Las empresas lo saben, y buscan comercio en las aplicaciones móvil, de hecho, hay que destacar que la mayoría de las aplicaciones analizadas tenían la misma estructura y mismo diseño, lo que quiere decir, que fue la misma empresa la que realizó la funcionalidad y diseño de dichas aplicaciones. Esto indica que hay un mercado en ese ámbito de la tecnología (diseño de aplicaciones móviles de ayuntamiento), lo que significa que, este proyecto podría resultar de utilidad para aquellas empresas que se dediquen al diseño de aplicaciones móviles de ayuntamiento.

Aunque la mayoría de ciudades, pueblos o municipios estén conectados a Internet, no todos poseen las mismas herramientas tecnológicas como una página web o una aplicación móvil. En realidad, no se da la importancia que debería tener a anunciarse en Internet, y seguro que cuando llegue alguien que exprima este negocio, va a ganar mucho dinero, hasta entonces, ese nicho sigue aun sin explotarse.

En este proyecto se ha propuesto dar el primer paso y realizar una estructura que sirva como guía para una futura aplicación móvil de ayuntamiento, y para ello se ha utilizado tanto contenido de informática, búsqueda y gestión de la información.

Ha resultado muy satisfactorio plasmar y utilizar conocimientos que se han visto y aprendido en el Máster en Gestión de la Información (MUGI). Además, se han conseguido nuevos conocimientos como el Scraping, la técnica novedosa para adquirir y almacenar información, gracias a la cual se ha podido recopilar la información de la tienda de GooglePlay.

En un principio se quería obtener la información, pero no se sabía que proceso iba a ser más adecuado para el proyecto, y el Scraping ha sido la elección perfecta, por su facilidad de aprender y su simple manejo.

Una vez cumplido el primer objetivo específico, surgieron dudas en cómo analizar la información, ya que su obtención había sido automática, el hacerlo manual iba a ser bastante costoso, pero al final, se optó por hacerlo manualmente y adquirir conocimientos que pueden ser útiles en un nuevo proyecto.

Lo mismo ocurrió con el modo de almacenar la información, si hacer las anotaciones en papel y bolígrafo, hacer imágenes y compararlas, etc. Para poder completar el segundo objetivo específico, se escogió guardarlas en una hoja de Excel. Aunque parezca una herramienta muy simple, todo lo contrario, es una herramienta que puede resultar muy compleja y, además, puede servir para muchas cosas, en este caso, almacenar información, visualizarlo correctamente y comparar funciones de forma sencilla.

10. BIBLIOGRAFÍA

2016. How People Use Their Devices, What Marketers Need to Know. Think with Google.

Disponible en:

<https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwi1nNvksNvUAhUJuhQKHbuCBIMQFgggMAA&url=https%3A%2F%2Fwww.thinkwithgoogle.com%2Fqs%2Fdocuments%2F276%2Ftwg-how-people-use-their-devices-2016.pdf&usg=AFQjCNFe1uIBDXIZLTa3fCDNDcge3WaDSQ> [Consulta junio de 2017]

2017. Wikipedia, La enciclopedia libre. Python. Disponible en:

<https://es.wikipedia.org/wiki/Python> [Consulta junio de 2017]

2017. Wikipedia, La enciclopedia libre. Web scraping. Disponible en:

https://es.wikipedia.org/wiki/Web_scraping [Consulta junio de 2017]

Página web, Curso de Python. Disponible en:

<https://sites.google.com/site/cursodepython/modulo-sys> [Consulta junio de 2017]

Anaya, Jesus, 2015. La librería requests de python. bc.py. Disponible en:

<https://pythonbc.com/blog/la-libreria-requests-de-python/> [Consulta junio de 2017]

2011. Crear, Leer y Modificar una Hoja Excel con Python. Ubuntu Life #WorldRevolution.

Disponible en: <https://ubuntulife.wordpress.com/2011/09/25/crear-leer-y-modificar-una-hoja-excel-con-python/> [Consulta junio de 2017]

2017. pprint - Data pretty printer. The Python Standard Library, Data Types. Disponible en:

<https://docs.python.org/3/library/pprint.html> [Consulta junio de 2017]

Pascual, Juan Antonio, 2015. Cómo instalar Ubuntu en un equipo con Windows. ComputerHoy.com . Disponible en: <http://computerhoy.com/paso-a-paso/software/como-instalar-ubuntu-equipo-windows-35619> [Consulta junio de 2017]

Moya, Ricardo, 2015. Scraping en Python (BeautifulSoup), con ejemplos. Jarroba.com. Disponible en: <https://jarroba.com/scraping-python-beautifulsoup-ejemplos/> [Consulta junio de 2017]

2015. Ayuntamiento de Cartagena. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=com.aytocartagena.android> [Consulta junio de 2017]

2016. Ayuntamiento de Mieres. Google Play. Disponible en: https://play.google.com/store/apps/details?id=com.app_aytomieres.layout [Consulta junio de 2017]

2017. App Valencia. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=es.valencia.lanzadera> [Consulta junio de 2017]

Triangle Postals S.L. Viajes y guías. 2017. Barcelona. Guía oficial. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=cat.triangle.bcn> [Consulta julio de 2017]

Madrid Destino Cultura Turismo y Negocio S.A. Viajes y guías. 2017. Guía Bienvenidos a Madrid. Google Play. Disponible en: https://play.google.com/store/apps/details?id=com.esmadrid.md_bam [Consulta julio de 2017]

Ayuntamiento de Sevilla Viajes y guías. 2017. Sevilla. Google Play. Disponible en:
<https://play.google.com/store/apps/details?id=es.sdos.visitasevillamobile&hl=es> [Consulta
julio de 2017]

ANEXO I – CÓDIGO DE EXTRACCIÓN DE LA INFORMACIÓN

```
#!/usr/bin/python

# Do not remove
GOOGLE_LOGIN = GOOGLE_PASSWORD = AUTH_TOKEN = None

import sys
import requests
import xlwt
from pprint import pprint
from bs4 import BeautifulSoup

from config import *
from googleplay import GooglePlayAPI
from helpers import sizeof_fmt, print_header_line, print_result_line

#Excel
wb = xlwt.Workbook()
ws = wb.add_sheet('A Test Sheet')
ws.col(0).width = 13 * 625
ws.col(1).width = 13 * 625
ws.col(2).width = 13 * 625
ws.col(3).width = 13 * 625
ws.col(4).width = 13 * 625
ws.col(5).width = 26 * 625
style = 'pattern: pattern solid, fore_colour yellow;'
fmt = xlwt.Style.easyxf("""borders: left thick, right thick, top thick, bottom thick;pattern: pattern solid, fore_colour yellow;""")
ws.row(0).write(0,"NOMBRE DE LA APLICACION",fmt)
ws.row(0).write(1,"CORREO ELECTRONICO",fmt)
ws.row(0).write(2,"NUMERO DE DESCARGAS",fmt)
ws.row(0).write(3,"ULTIMA ACTUALIZACION",fmt)
ws.row(0).write(4,"NUM DE ESTRELLAS (MAX 5)",fmt)
ws.row(0).write(5,"ENLACE DE LA APLICACION",fmt)

if (len(sys.argv) < 2):
    print "Usage: %s request [nb_results] [offset]" % sys.argv[0]
    print "Search for an app."
    print "If request contains a space, don't forget to surround it with \"\"\"
    sys.exit(0)

request = sys.argv[1]
nb_res = None
offset = None

if (len(sys.argv) >= 3):
    nb_res = int(sys.argv[2])

if (len(sys.argv) >= 4):
    offset = int(sys.argv[3])

api = GooglePlayAPI(ANDROID_ID)
api.login(GOOGLE_LOGIN, GOOGLE_PASSWORD, AUTH_TOKEN)

try:
    message = api.search(request, nb_res, offset)

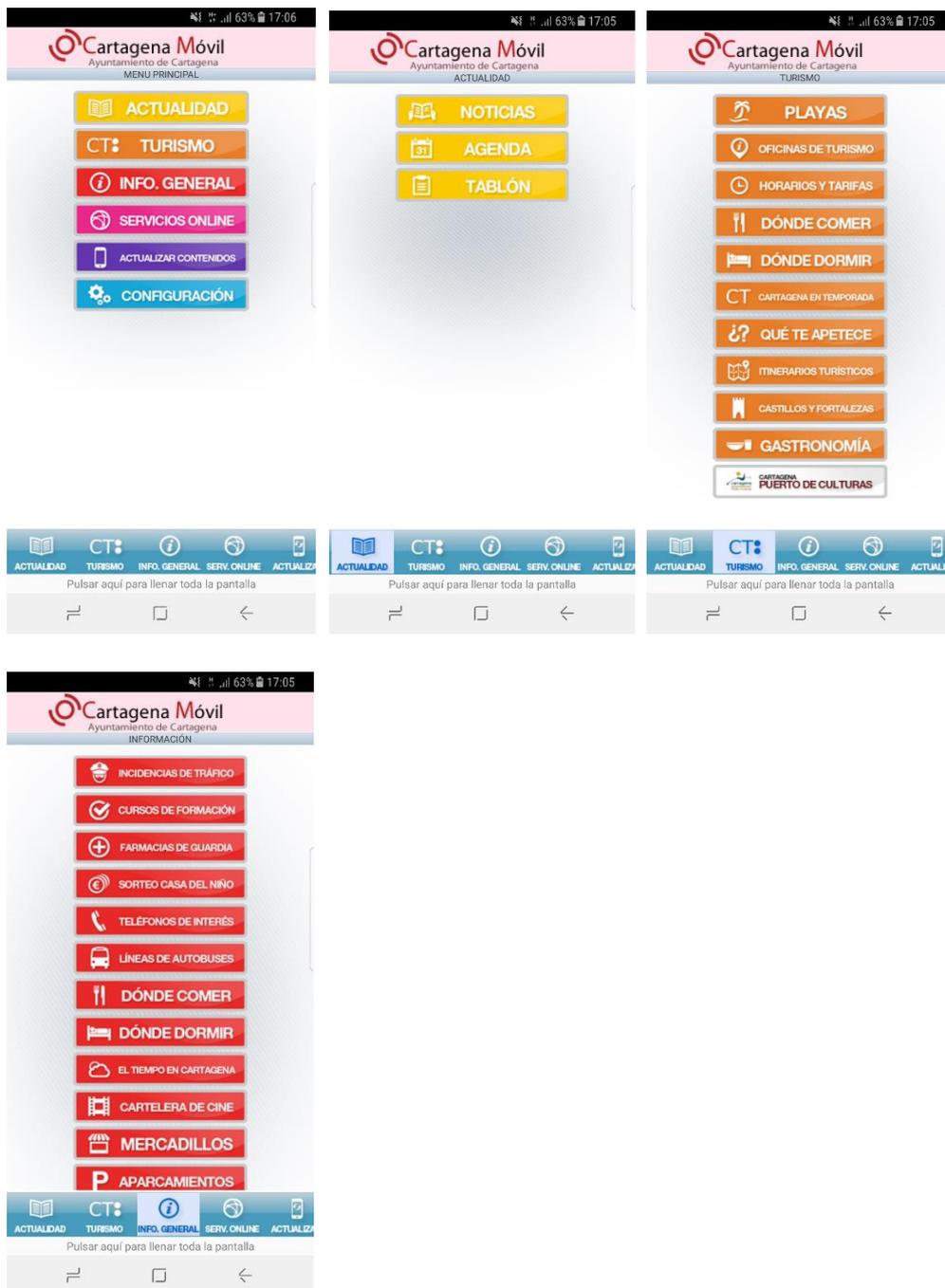
except:
    print "Error: something went wrong. Maybe the nb_res you specified was too big?"
    sys.exit(1)

apps = []
mails = []
descargas = []
actualizaciones = []
ranking = []
enlaces = []
#print_header_line()
doc = message.doc[0]
#indice = 0
for c in doc.child:
    #print_result_line(c)
    shareUrl = c.shareUrl
    title = c.title
    url = shareUrl
    req = requests.get(url)
    statusCode = req.status_code
    aux = []
    apps.append(c.title)
    descargas.append(c.details.appDetails.numDownloads)
    actualizaciones.append(c.details.appDetails.uploadDate)
    ranking.append(c.aggregateRating.starRating)
    enlaces.append(c.shareUrl)
    if statusCode == 200:
        html = BeautifulSoup(req.text)
        correo = html.find_all('a',{'class':'dev-link'})
        for j,correo in enumerate(correo):
            href = correo.get('href')
            if not href.find("mailto"):
                mails.append(href)
            if len(apps) > len(mails):
                mails.append("no tiene")
# indice = indice + 1
```

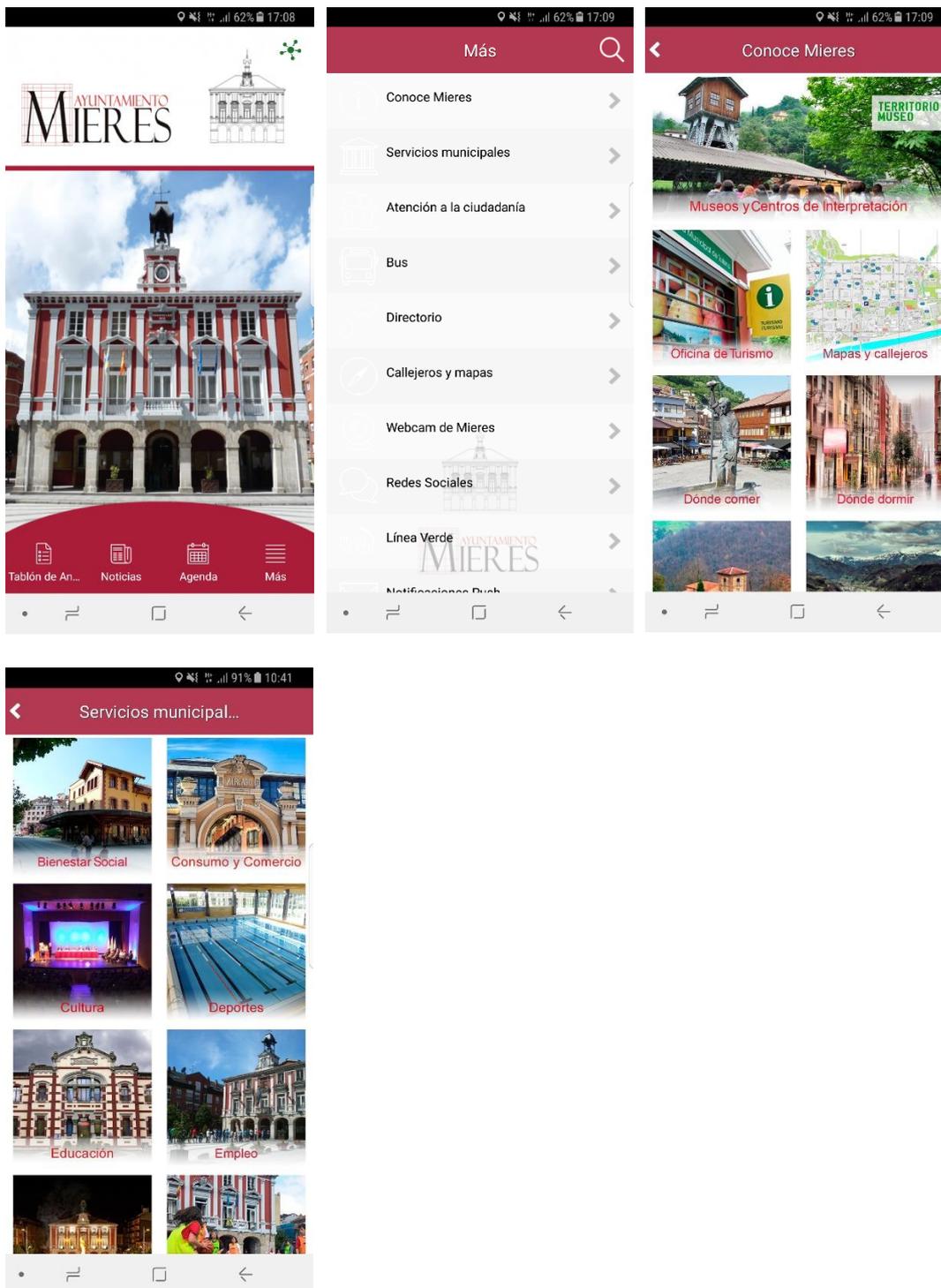
```
buscar = "mailto:"
reemplazar = ""
for t in range(len(mails)):
    mails[t] = mails[t].replace(buscar,reemplazar)
#print len(apps)
#print len(mails)
for s in range(len(apps)):
    (ws.row(s+1)).write(0,apps[s])
    (ws.row(s+1)).write(1,mails[s])
    (ws.row(s+1)).write(2,descargas[s])
    (ws.row(s+1)).write(3,actualizaciones[s])
    (ws.row(s+1)).write(4,ranking[s])
    (ws.row(s+1)).write(5,enlaces[s])

wb.save('Correos.xls')
print "Hecho. OK!"
```

ANEXO II – APLICACIÓN DE CARTAGENA



ANEXO III – APLICACIÓN DE MIERES



ANEXO III – APLICACIÓN DE VALENCIA

