



CICLO FORMATIVO DE GRADO SUPERIOR
DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROYECTO FIN DE CICLO

WHERE I PARK

**Daniel Alonso Casilda
Ignacio López Jiménez
Andrea Muñoz Hidalgo**

CURSO 2019-20

TÍTULO: Where I Park 1.0

AUTORES: Daniel Alonso Casilda

Ignacio López Jiménez

Andrea Muñoz Hidalgo

TUTOR DEL PROYECTO: Ernesto Ramiro Córdoba

FECHA DE LECTURA: 8 de junio de 2020

En Madrid

Ernesto Ramiro Córdoba
Tutor del PFC

RESUMEN:

La movilidad personal en los tiempos en los que nos encontramos es cada día más una necesidad impuesta y no una elección personal.

Nos movemos más rápido y más lejos que nunca gracias a tecnologías que tenemos al alcance de nuestra mano. Esto hace que nos desplacemos a muchos lugares en un día, y en ocasiones cuando aparcamos, podemos despistarnos y no prestar atención a donde hemos aparcado nuestro vehículo, los lugares son masivos, repetitivos, demasiado iguales...

Encontramos gran cantidad de aplicaciones que nos guardan una ubicación al estilo de WhatsApp, Telegram... pero ninguna de ellas está optimizada para guardar, enviar y consultar ¿dónde dejé mi coche?

En este punto entra "Where I Park". Rápido, sencillo, eficaz y sin rodeos.

El usuario puede sin problemas guardar su ubicación, compartirla y anotar indicaciones.

En el momento en el que tenga el usuario que consultar donde dejó su vehículo no tendrá que abrir una aplicación de mensajería, abrir el chat de conversación donde se envió la ubicación, abrir el maps, leer una nota...todo esto lo soluciona "Where I Park".

El usuario abrirá la aplicación y automáticamente aparecerá en pantalla el punto exacto donde aparcó, y una anotación si necesita recordar algo, rápido y limpio.

Queremos saber dónde está nuestro vehículo, no hace falta tener una experiencia espiritual con nuestro móvil.

ABSTRACT:

Personal mobility in these times has increasingly become an imposed necessity rather than a personal choice.

We move faster and further than ever thanks to technologies within reach. This makes us move to many places in one day, and sometimes when we park, we can get confused and not pay attention to where we have parked our vehicle, the places are massive, repetitive, too similar...

We can find a large number of applications that are able to save our location like WhatsApp, Telegram... but none of them is optimized to save, send, and consult where we have left our car.

At this point comes "Where I Park". Fast, simple, effective, and direct.

The user can easily save their location, share it and write down notes.

In the moment in which the user needs to consult where it has left its car, the user won't have to open any messaging application, open the chat where the location was sent, open google maps, read the notes... "Where I Park" solves all of this.

The user will open the application and the exact point where he/she has park will automatically appear on the screen along with a note if necessary, fast, and clean.

We want to know where our vehicle is, it's not necessary to have a spiritual experience with our mobile.

AGRADECIMIENTOS

A todos nuestros profesores que nos han ayudado estos dos años, en especial a Pilar Martín Gómez, que con su dedicación y su paciencia ha conseguido enseñarnos todas las herramientas necesarias para poder comenzar nuestra carrera en esta profesión, y ha conseguido hacer de nosotros pequeños programadores con muchas ganas de seguir creciendo y aprendiendo.

También queremos agradecer a Pilar su implicación con nosotros y su atención, tanto dentro como fuera de su horario de trabajo, para todas esas pequeñas o grandes dudas que han ido surgiendo durante este tiempo, no solo con su asignatura, sino con todo lo referente a nuestro futuro profesional.

Queremos transmitir también un agradecimiento muy especial a Ernesto, cuyo entusiasmo, atención y buen gusto han sido cruciales para este proyecto, siendo la persona idónea para enseñarnos la importancia del diseño, cómo ver más allá del código de una aplicación y el alcance que esta puede proyectar.

A nuestra familia y amigos, especialmente a Iván González y Gonzalo García, dos personas que, además de grandes profesionales, son grandes amigos que nos han ofrecido su inestimable ayuda y apoyo durante estos casi tres meses de proyecto para poder resolver todos aquellos problemas que han ido surgiendo y han sabido orientarnos para trabajar de manera más eficaz y profesional.

Por último, queremos agradecer el ecosistema de enseñanza que nos brinda la Universidad Europea para facilitarnos lograr nuestros objetivos.

Asimismo, hay que recalcar que la obra que se expone a continuación ha sido posible por la entera dedicación y compromiso de los integrantes de este equipo, Daniel Alonso Casilda, Ignacio López Jiménez y Andrea Muñoz Hidalgo.

Esta obra se distribuye bajo una licencia Creative Commons.

Se permite la copia, distribución, uso y comunicación de la obra si se respetan las siguientes condiciones:

- Se debe reconocer explícitamente la autoría de la obra incluyendo esta nota y su enlace.
- La copia será literal y completa
- No se podrá hacer uso de los derechos permitidos con fines comerciales, salvo permiso expreso de los autores.

El texto precedente no es la licencia completa sino una nota orientativa de la licencia

original completa (jurídicamente válida) que puede encontrarse en:

<http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es>

INTRODUCCIÓN	7
Investigación previa sobre la práctica	7
Objetivos	7
Motivación	8
Antecedentes	9
DESARROLLO DEL PROYECTO	10
Herramientas generales del proyecto	10
Sistemas de control de versiones	11
Herramientas de Ofimática	11
Software de organización de proyectos	12
Herramientas de comunicación	12
Herramientas de diseño	13
Herramientas de desarrollo	14
Herramienta de información de sistema	15
Planificación	15
Punto de partida	17
Desarrollo del MockUp	17
Desarrollo del código	20
Sistemas de control de versiones	20
Instalación y configuración	20
IDE de programación	20
instalación y configuración	20
Código destacado	21
Clase PreferencesHelper	21
savePreferences()	21
loadPreferences()	21
deletePreferences()	21
Carga de launcher inicial	22
launcher.xml	22
styles.xml	22
AndroidManifest.xml	23
Clase UbicationModel	23
Clase MainActivity	23
configureDrawer()	23
shareLocation()	24
configureActivity()	25
MapFragment	26
onMapReady()	26

onRequestPermissionsResult()	27
fetchLastLocation()	28
addNotes()	28
bitmapDescriptorFromVector()	29
RESULTADOS Y VALIDACIÓN	30
Diagrama de clases de la aplicación	30
Consumo de recursos	30
Terminales de prueba	30
Recursos generales	34
Consumo de CPU	35
Consumo de RAM	35
Consumo de datos	35
Consumo de energía	36
CONCLUSIONES	37
Innovación	37
Trabajo futuro	37
BIBLIOGRAFÍA Y WEBGRAFÍA	38
Google Maps API	38
Comunicación Activity/Fragment	41
Navigation Drawer + Map Fragment	42
Toolbar personalizada	44
Launcher / Splash Screen	45
Shared Preferences	46
Comunicación entre apps	47
JavaDoc	47
Documentación para la memoria	48

INTRODUCCIÓN

Investigación previa sobre la práctica

Los objetivos de desarrollar una aplicación Android como PFC engloba diferentes ámbitos:

1 - Plasmar una metodología de trabajo de proyecto basada en sistemas ágiles Scrum y Kanban, con reuniones diarias del equipo en las que analizaremos día a día los hitos pasados en la implementación de funciones, pequeños sprints internos, todo ello enfocado a que cada uno de los integrantes del equipo se “contagie” de los conocimientos de cada uno de los integrantes, haciendo la evolución del proyecto un ecosistema enriquecedor para todas las partes.

2 - Combinar los conocimientos técnicos en el desarrollo adquiridos durante el ciclo formativo y evolucionarlos a un entorno profesional real de producción, por ejemplo, implementar correctamente una pantalla de bienvenida, manejar datos internos en la aplicación mediante SharedPreferences, compartir datos entre aplicaciones, hacer un entorno ligero y rápido.

Objetivos

Conocer el funcionamiento de herramientas para desarrollar el proyecto, siendo estas:

- IDE de desarrollo: Android Studio 3.6.3
- Software de gestión de proyectos: Trello
- Servicios Ofimática Cloud: G Suite
- Gestor de reuniones telemáticas: Skype, WhatsApp
- Constructor de Interfaz gráfica: Adobe XD, Gimp, Inkscape, NotebookCast
- Mensajería para la organización: Slack
- Sistema de control de versiones: Git
- Clientes GUI Git: SourceTree
- Repositorio de código: GitLab, GitHub
- Lenguaje de programación: Java

- Gestores de escritorio remoto: TeamViewer, AnyDesk

A nivel de investigación e implementación de código los objetivos son el desarrollo basado principalmente con:

- Implementación y uso SharedPreferences
 - Mediante este sistema podremos guardar datos necesarios de la aplicación a modo de base de datos, siendo estos guardados de una manera privada y segura. El usuario será capaz de modificarlos, crearlos, borrarlos...
- Launcher Theme previa a la carga de la app
 - Con este sistema de carga inicial se evita las pantallas de carga por defecto (normalmente una pantalla blanca), que carga el sistema operativo, de este modo la aplicación tiene un aspecto más amigable al usuario, ya que la app lo primero que hará antes de empezar a cargarse será mostrar una pantalla de bienvenida, de este modo la experiencia de usuario será la misma independientemente de la potencia del dispositivo, carga del sistema en el momento de abrir la app...
- Google Maps SDK for Android API v2 como api principal
 - Ya que la aplicación se basa en geolocalización y muestra, modificación de los datos, la API a su última versión estable establece las herramientas necesarias para que la experiencia de usuario sea rápida y eficaz.
- Send Locations a diferentes app de mensajería
 - Compartir las coordenadas de geolocalización entre diferentes apps que tenga instalada el usuario.

Motivación

Comprobamos que las aplicaciones que comúnmente las personas usan a este efecto, son aplicaciones que no están destinadas a ello, muy complejas, por lo que se hace necesario WIP.

En la mente del equipo se centra en desarrollar los elementos más usados en las aplicaciones modernas, tales como los descritos anteriormente de compartir, uso de SharedPreferences, Launcher Theme sin timings.

En segundo bloque de motivaciones aborda desde la visión general de desarrollo del producto, desde una primera instancia típica con un brainstorming inicial, el desarrollo de todo, hasta la publicación del producto

final en las plataformas de descarga.

Antecedentes

Actualmente en el mercado contamos con gran cantidad de aplicaciones basadas en las soluciones que nos proporcionan diferentes apis de desarrollo centradas en mapas y ubicación, siendo sin duda una de las más usadas la API de Google Maps.

Esta API ofrece gran cantidad de servicios y facilidades de integración a los desarrolladores de aplicaciones, lo que hace que proliferen gran cantidad de productos que en sus inicios eran muy básicos.

Los desarrolladores de estas aplicaciones básicas fueron poco a poco, incorporando en sus productos las diferentes opciones que iban apareciendo en las sucesivas revisiones por parte de Google de su API, en un afán de acaparar mercado, centrándose en la visión comercial de que, si el producto tenía más opciones, los usuarios optarán por tener esa app instalada y no la de la competencia.

Este modelo de producción tuvo sentido durante el tiempo en el que los dispositivos móviles tenían un coste elevado, las memorias en estado sólido no eran tan baratas como ahora, los procesadores demandaban mucho consumo de energía (tenían litografías muy grandes comparadas con hoy en día), las baterías tenían menos amperaje...en definitiva, se comenzaba un proceso de miniaturización, mejora de rendimiento y recursos similar a los computadores portátiles con sistemas operativos de gran repercusión, pero a diferencia del proceso de los PC, esto no se ha llevado a cabo en 25 años sino en 12 años.

Esta reducción en los tiempos, en los que los dispositivos móviles han sobrepasado los casos de uso a los computadores portátiles, ha hecho que lo que era válido hace 5 años, se haya quedado obsoleto en la actualidad, por lo que a lo que refiere a las aplicaciones con funciones de geolocalización, con muchísimas opciones diferentes, sea un tipo de producto que el usuario empieza a dejar de demandar.

Se está empezando a crear un usuario nuevo, el cual prefiere pequeñas aplicaciones que le resuelvan a la perfección un pequeño problema, sin tener que investigar en grandes aplicaciones.

Es en este nicho de mercado donde entran aplicaciones como WIP “**guardar rápidamente y sin rodeos una ubicación donde estoy, y recordarla igualmente rápido**”.



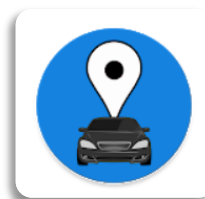
ParKing - ¿Dónde está mi carro? - Automático



Map Marker



Encuentra el coche aparcado - Dónde dejé el auto



Encontrar mi Coche Aparcado

Existen muy pocas aplicaciones que resuelvan guardar la ubicación del aparcamiento de modo rápido y sencillo. Analizando estas Apps WIP mejora la velocidad, rendimiento y consumo de recursos al funcionar de un modo más directo y eficaz.

DESARROLLO DEL PROYECTO

Herramientas generales del proyecto

Con el fin de proporcionar la máxima flexibilidad a la hora de el desarrollo del proyecto se considera necesario que las herramientas sean gratuitas, y que se puedan implementar en cualquier sistema operativo mayoritario (Windows, Mac, Linux) pudiendo así ser obtenidas por cualquier desarrollador, por ello en la descripción de cada una de las herramientas se proporciona el enlace oficial del software o recurso web cloud.

Compatibilidades

La aplicación se desarrollará para que esté perfectamente implementada en versiones de Android 4.0.3 con el API 15 (IceCreamSandwich ICS) en adelante ya que a fecha de 4 de abril de 2020 la Android Studio nos garantiza la compatibilidad con el 100% de dispositivos.

Sistemas de control de versiones

- **Software de control de versiones Git**

Con él se podrá llevar el registro de los cambios en archivos de cada integrante del equipo y coordinar el trabajo que estos realizan sobre archivos compartidos. Actualmente es el estándar profesional en el software de control de versiones. Se trabajará con la versión 2.26.1

<https://git-scm.com/>



Logo de Git

- **Repositorio de código en GitLab**

Se acuerda GitLab con el fin de conocer y usar nuevos repositorios usados en el entorno profesional.

<https://about.gitlab.com/>

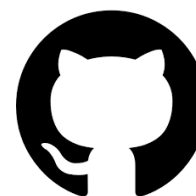


Logo De GitLab

- **Repositorio de código en Github**

Se opta por crear un repositorio clonado de GitLab a GitHub como repositorio oficial del proyecto con el fin de habilitar una versión a la Universidad.

<https://github.com/>



Logo de GitHub

- **SourceTree**

Cliente GUI para manejar repositorios git, ya que la fiabilidad de esta herramienta hace posible una visualización práctica y rápida de el estado de los repositorios. Se acuerda funcionar con la versión 3.3.8

<https://www.sourcetreeapp.com/>



Logo de SourceTree

Herramientas de Ofimática

- **Google Suite**

Nos permitirá sacar el máximo partido a la suite de ofimática en cloud con lo que ello conlleva en el trabajo en equipo a tiempo real.

<https://www.google.es/intl/es/docs/about/>



Logo de G Suite

Software de organización de proyectos

- **Trello**

Nos dará capacidad de crear tableros en los que vemos claramente qué tareas están pendientes de empezar, cuales en proceso y cuales terminadas, es una buena herramienta de simulación en la gestión del proyecto que puede asemejar un tablero Kanban.

<https://trello.com>



Logo de Trello

Herramientas de comunicación

- **Skype**

Como herramienta de reunión a distancia, tiene la ventaja respecto a la competencia de herramientas en cuanto a la estabilidad del sistema, calidad en el audio / vídeo sin cortes así como la posibilidad de compartir código de una manera visual muy rápida.

<https://www.skype.com>



Logo de Skype

- **Slack**

Como comunicación externa al equipo de desarrollo, ya que es una herramienta muy usada en el entorno profesional con innumerables aplicaciones que poder conectar orientadas a la productividad, diseño, estrategia...una fuente importante de descubrimiento de nuevas opciones.

<https://slack.com>



Logo de Slack

- **WhatsApp**

Herramienta de comunicación inmediata alternativa a Skype.

<https://www.whatsapp.com>



Logo WhatsApp

- **TeamViewer**

Uniéndose las versiones de Android + Windows/Mac permite compartir la pantalla en directo del terminal a otros usuarios, de este modo se genera una alternativa a el visionado del diseño de la app al emulador de Android Studio.

<https://www.teamviewer.com>



Logo TeamViewer

- **AnyDesk**

Herramienta de escritorio remoto rápida y ágil como alternativa a TeamViewer.

<https://anydesk.com/>



Logo AnyDesk

Herramientas de diseño

- **Adobe XD**

Como herramienta de prototipado ya que respecto a la competencia la calidad de los resultados respecto al tiempo empleado ofrece una calidad de trabajo visualmente muy superior.

Esta herramienta ofrece la capacidad de compartir el trabajo de un modo profesional y rápido al destinatario, pudiendo ofrecer un feedback prácticamente inmediato.

<https://www.adobe.com/products/xd.html>



Logo de Adobe XD

- **Gimp**

Como editor gráfico de mapa de bits para la creación de logos, botones, iconos, punteros...

<https://www.gimp.org/>



Logo de Gimp

- **InkScape**

Como editor de imágenes vectoriales para los recursos que necesiten una adaptación perfecta a la resolución de la pantalla de los diferentes dispositivos.

<https://inkscape.org/>



Logo de InkScape

- **NoteBookCast**

Como pizarra virtual para el comienzo en el desarrollo del MockUp

<https://www.notebookcast.com/>

NoteBookCast

Logo de NoteBookCast

Herramientas de desarrollo

- **Java**

Como lenguaje de programación de la aplicación, ya que este es el lenguaje más usado mes a mes según la lista TIOBE y ofrece una perfecta integración en el entorno de desarrollo Android. La cantidad de documentación que existe para resolver problemas es muy amplio y accesible, por lo que la facilidad a la hora de trabajar bajo este lenguaje hace que sea una ventaja respecto al resto.

<https://www.oracle.com/java/>



Logo de Java

- **Android Studio**

Como IDE de programación es el más apropiado para implementar una aplicación en Android. Es la plataforma de desarrollo oficial de Google, para hacer aplicaciones al sistema operativo de Google e incluso da la capacidad de hacer el desarrollo de las aplicaciones en el lenguaje oficial de Google (Kotlin), por lo que con todo esto nos garantizamos un ecosistema perfectamente implementado para la plataforma. Con el fin de unificar el IDE en todo el equipo de trabajo ve trabajará en la última versión disponible a fecha de 1 de Abril de 2020.

<https://developer.android.com/studio/index.html>



Android Studio 3.6.3 1
de Abril 2020

- **Maps SDK for Android V2**

Nos permitirá agregar mapas basados en datos de Google Maps a nuestra aplicación. La API controla automáticamente el acceso a los servidores de Google Maps, la descarga de datos, la visualización de mapas y la respuesta a los gestos del mapa. También podremos utilizar las llamadas a la API para agregar marcadores, polígonos y superposiciones a un mapa básico, así como



para cambiar la vista que tenga el usuario de un área del mapa en particular. Estos objetos proporcionan información adicional de las ubicaciones en el mapa y permiten la interacción del usuario con este.

<https://developers.google.com/maps/documentation/android-sdk/intro>

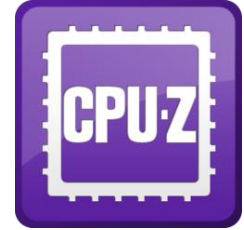
Logo Google Maps

Herramienta de información de sistema

- **CPU-Z**

Con esta herramienta podremos tener una información detallada del dispositivo android en el que vamos a ejecutar la aplicación terminada.

https://play.google.com/store/apps/details?id=com.cpu_z&gl=ES

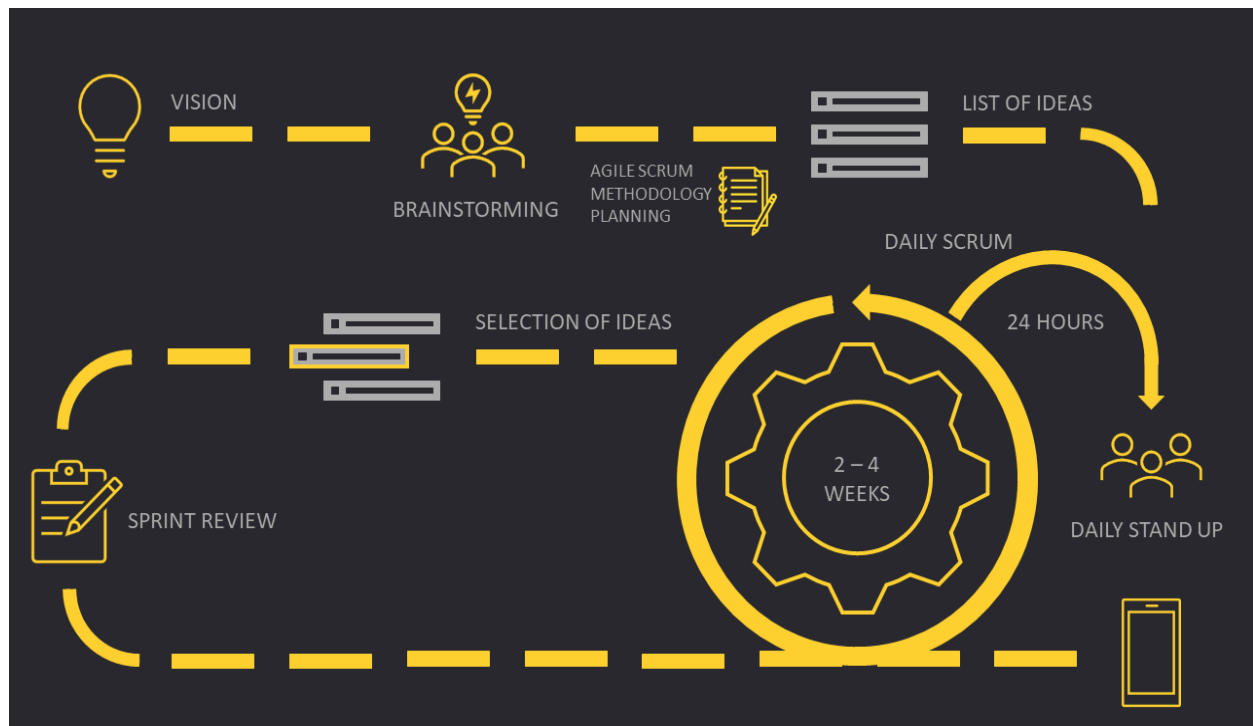


Logo de CPU-Z

Planificación

Se planifica el trabajo en sesiones diarias de aproximadamente seis horas en las que la vía de comunicación continua es mediante Skype, de este modo se rompe el aislamiento del trabajo, fomentando un flujo constante de desarrollo en equipo.

De este modo y con este modelo de desarrollo se intenta simular un método de trabajo dentro del cual los desarrolladores de software se implementen todos dentro del mismo departamento.



Modelo de trabajo de WIP

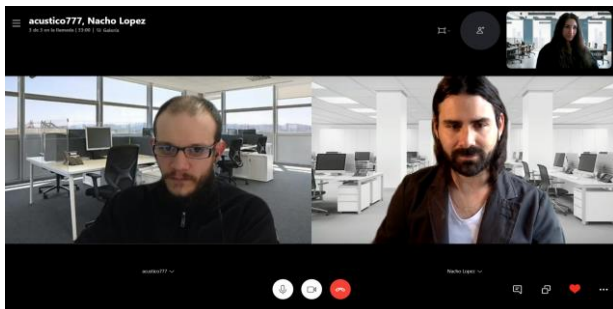
Función práctica de la planificación diaria:

- Visión del trabajo efectuado en la sesión del día anterior.
- Puesta en común de ideas para llevar a cabo como implementar la lista de tareas a ejecutar ese día.
- Ejecución de las tareas a implementar por parte de todo el equipo (departamento)
- Testeo de estabilidad funcionamiento
- Nueva lista de tareas teniendo en cuenta el nuevo código implementado teniendo en cuenta (SI / NO):
 - Se ha llegado al objetivo
 - Se ha implementado el objetivo tal y como se había previsto
 - Se mantiene la velocidad estimada en la ejecución de proyecto para la entrega final.

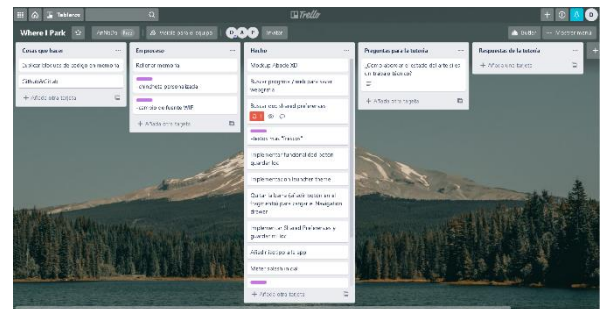
Punto de partida

Durante la primera sesión de brainstorming se usa Skype como herramienta de comunicación para lo cual se crea una sala de reuniones para los integrantes del equipo, que se mantendrá durante todo el desarrollo del proyecto.

Trello, creando un tablero de desarrollo que se mantendrá durante todo el proyecto. como tablero de ideas para gestionar la fusión de estas y hacer de ello un proyecto con forma determinada.



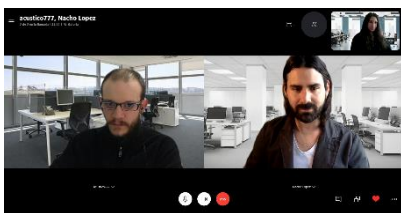
Equipo de trabajo



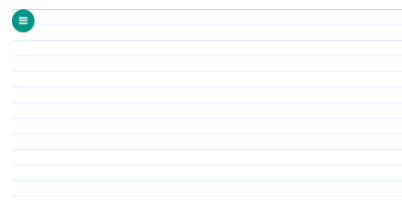
Ejemplo de tablero en Trello

Desarrollo del MockUp

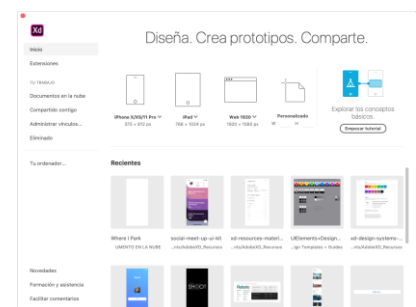
Unificando Skype como herramienta de comunicación y NoteBookCast + AdobeXD como herramientas de desarrollo.



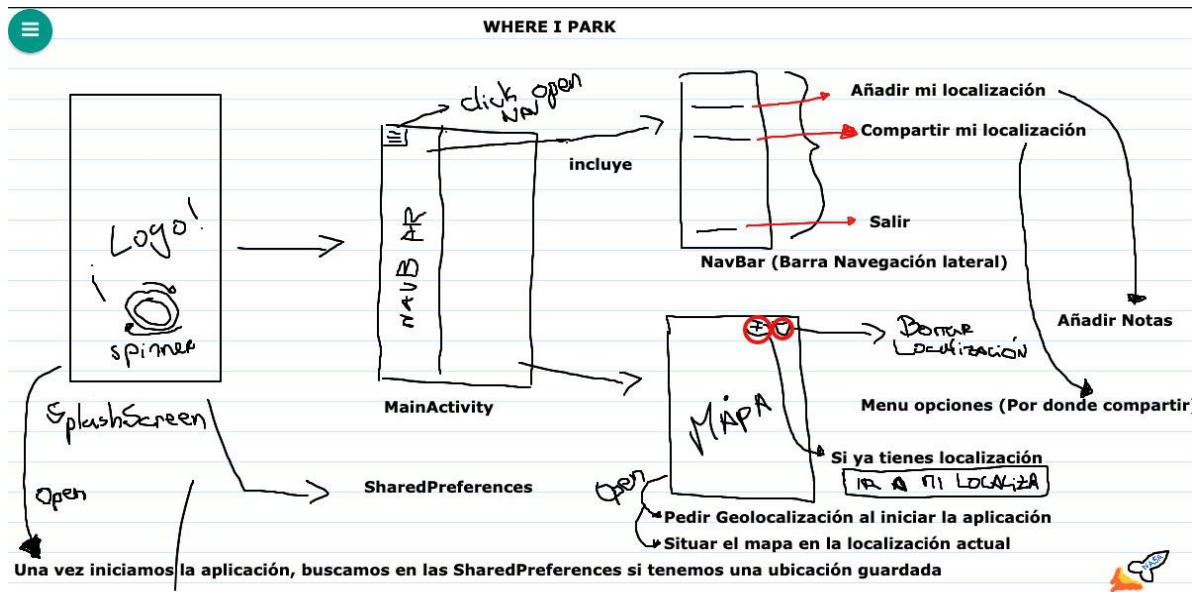
Equipo de trabajo



Pizarra en blanco de
NoteBookCast

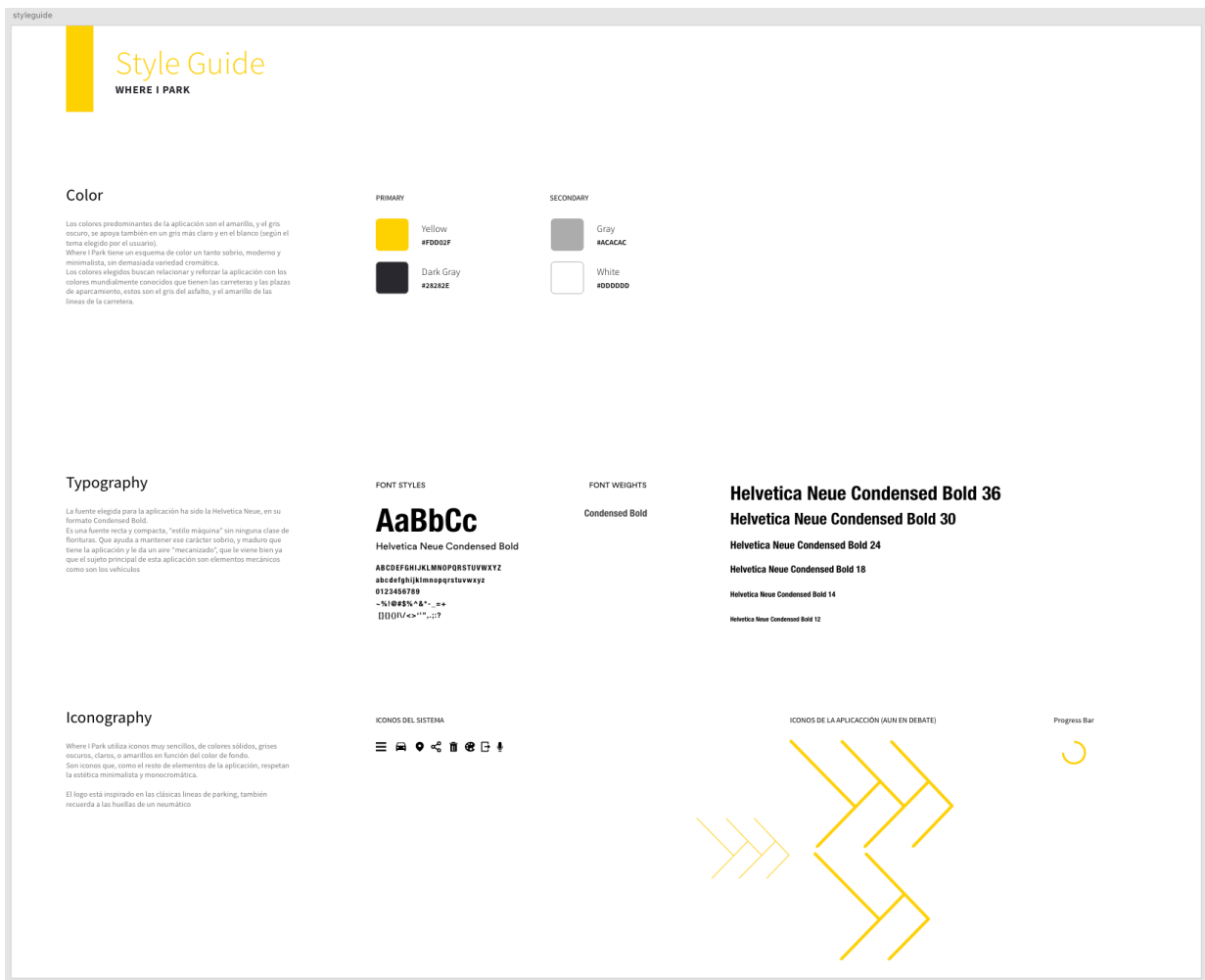


Comienzo de proyecto con
Adobe XD

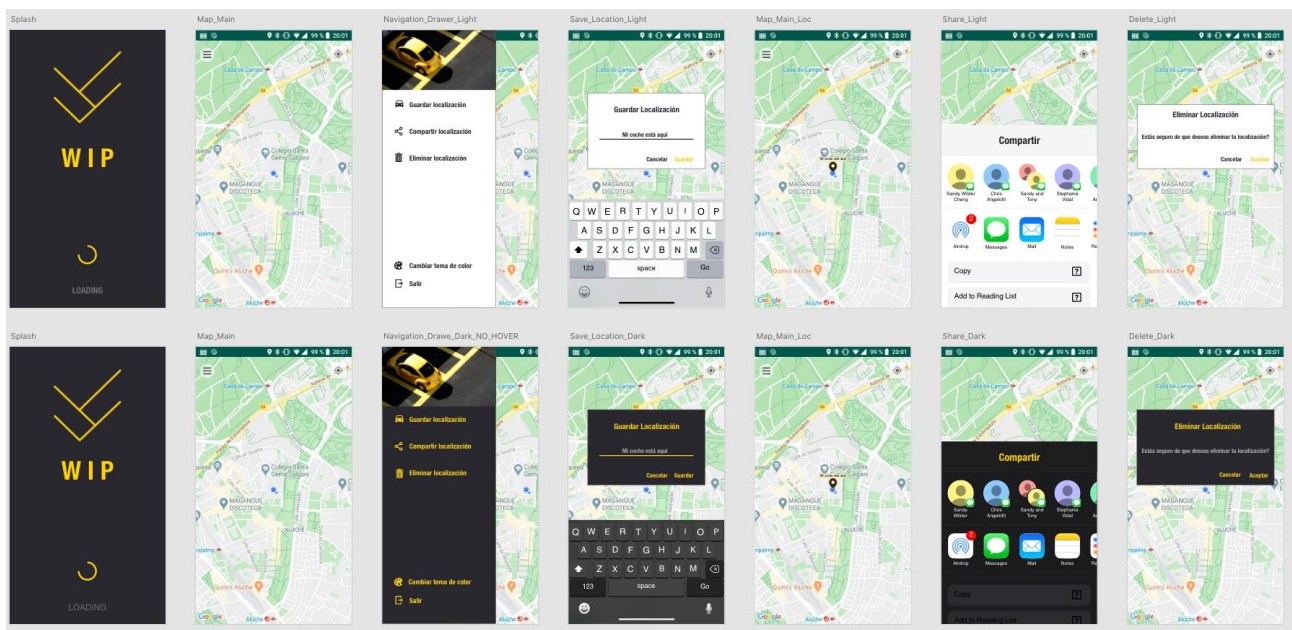


Primera versión de MockUp con NoteBookCast

Una vez acordado la última versión de mockUp se implementa el desarrollo junto a la guía de estilo mediante AdobeXD.



Guía de estilo con Adobe XD



MockUp finalizado con Adobe XD

Desarrollo del código

Basándonos en la metodología de trabajo previa que hemos llevado a cabo se implementan las herramientas de desarrollo:

Sistemas de control de versiones Instalación y configuración

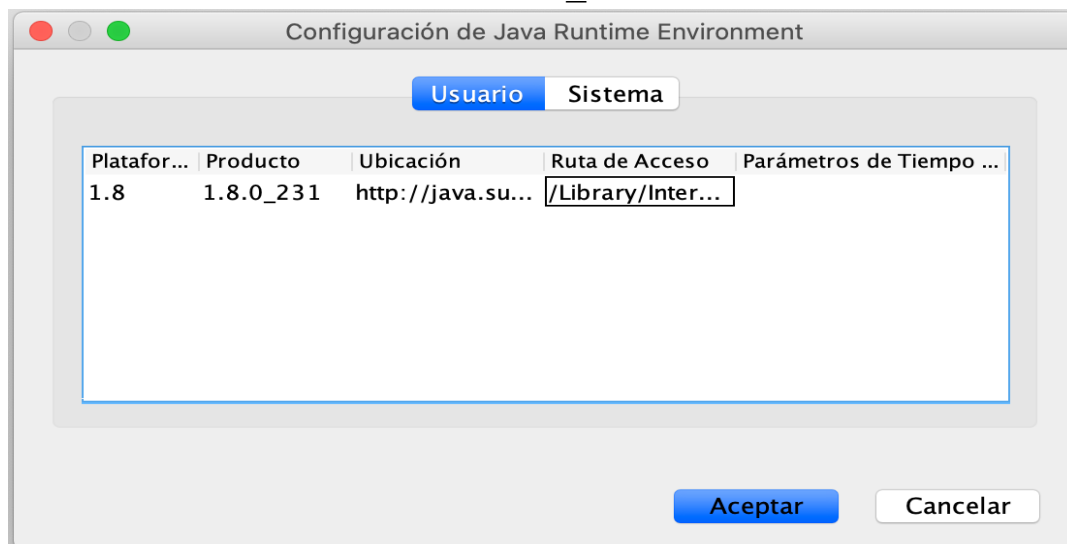
- Instalación de Git a última versión disponible siendo esta la 2.26.1
Creación de Repositorios en GitHub y GitLab
- Instalación de SourceTree última versión a 3.3.8
- Unión de repositorios mediante SourceTree

IDE de programación instalación y configuración

- Instalación de Android Studio version 3.6.3
- Actualización de dependencias a Gradle 3.6.3

```
9  ▶  dependencies {  
10      classpath 'com.android.tools.build:gradle:3.6.3'  
11  }
```

- Instalación a versión de Java 1.8.0_231



Código destacado

En este apartado se detalla las partes del código y clases más destacadas que hacen característico el funcionamiento de la aplicación.

Clase PreferencesHelper

Funciona como una base de datos interna a cada dispositivo con funcionalidad de tipo clave-valor. En ella guardaremos, cargaremos o borrarremos los datos referentes a nuestro modelo.

savePreferences()

Método utilizado para guardar preferencias.

```
public static void savePreferences(Context context, UbicationModel
model)
    SharedPreferences preferences =
context.getSharedPreferences(UBICATIONS, context.MODE_PRIVATE);
    SharedPreferences.Editor editor = preferences.edit();

    editor.putFloat(LATITUDE, (float)model.getLatitude());
    editor.putFloat(LONGITUDE, (float)model.getLongitude());
    editor.putString(NOTES, model.getNotes());
    editor.apply();
}
```

Listado 1: método savePreferences()

loadPreferences()

Método utilizado para cargar preferencias.

```
public static UbicationModel loadPreferences(Context context) {
    SharedPreferences preferences =
context.getSharedPreferences(UBICATIONS, context.MODE_PRIVATE);

    float fLatitude = preferences.getFloat(LATITUDE, DEF_LAT);
    float fLongitude = preferences.getFloat(LONGITUDE, DEF_LNG);
    String sMsg = preferences.getString(NOTES, DEF_NOTES);

    return new UbicationModel(fLatitude, fLongitude, sMsg);
}
```

Listado 2: método loadPreferences()

deletePreferences()

Método utilizado para borrar preferencias.

```

    public static void deletePreferences(Context context) {
        SharedPreferences preferences =
context.getSharedPreferences(UBICATIONS, context.MODE_PRIVATE);
        SharedPreferences.Editor editor = preferences.edit();
        editor.clear();
        editor.apply();
    }
}

```

Listado 3: método deletePreferences()

Carga de launcher inicial

Sustituye la pantalla de carga blanca inicial por una pantalla personalizada. Mediante un estilo accedemos al xml en el que se definen sus características, y lo llamamos desde el Manifest.

launcher.xml

```

<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:opacity="opaque">

    <item android:drawable="@color/darkGrey"/>

    <item>
    <bitmap
        android:src="@drawable/wip_launcher"
        android:gravity="center" />
    </item>

</layer-list>

```

Listado 4: launcher.xml

styles.xml

```

<style name="AppTheme.Launcher">
    <item name="android:windowBackground">@drawable/launcher</item>
</style>

```

Listado 5: estilo del launcher

AndroidManifest.xml

```
<application android:theme="@style/AppTheme.Launcher">
```

Listado 6: llamada al estilo desde el Manifest

Clase UbicationModel

Clase en la cual se definen los atributos necesarios, su constructor y los métodos get() y set(). Representa el objeto que manejaremos a lo largo de toda la aplicación.

```
public UbicationModel(double latitude, double longitude, String notes) {  
    this.latitude = latitude;  
    this.longitude = longitude;  
    this.notes = notes;  
}
```

Listado 7: constructor de la clase UbicationModel

Clase MainActivity

Clase principal de la aplicación. Encargada de lanzar tanto el fragmento del mapa como el navigation drawer.

configureDrawer()

Método que se encarga de configurar el navigationDrawer y la escucha de todos sus elementos.


```

private void configureDrawer() {
    final NavigationView navView = findViewById(R.id.nav_view);
    navView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem menuItem)
    {
        int id = menuItem.getItemId();

        if (id == R.id.addLoc) {
            addLocation();

        } else if (id == R.id.shareLoc) {
            shareLocation();

        } else if (id == R.id.delete) {
            deleteLoc();
            Toast.makeText(MainActivity.this, R.string.delete_ok,
Toast.LENGTH_SHORT).show();

        } else if (id == R.id.llExit) {
            System.exit(0);
        }
        return true;
    }
});
}

```

Listado 8: método configureDrawer()

shareLocation()

Método encargado de compartir la localización. Recupera el modelo del SharedPreferences y en función de lo obtenido nos muestra un AlertDialog o comparte la localización a través de un chooser.

```

private void shareLocation() {
    context = getApplicationContext();
    model = PreferencesHelper.loadPreferences(context);

    if (model.getLongitude() == 0 && model.getLatitude() == 0){
        AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this, R.style.AlertDialogStyle);
        builder.setTitle(R.string.share_loc_dialog_title);
        builder.setMessage(R.string.share_loc_dialog_msg);
        builder.setPositiveButton(R.string.accept, null);
        AlertDialog dialog = builder.create();
        dialog.show();

    } else {
        Intent i = new Intent(Intent.ACTION_SEND);
        i.setType("text/plain");
        i.putExtra(Intent.EXTRA_TEXT, model.getNotes() + "
\nhttp://maps.google.com/maps?q="
                + model.getLatitude() + "," + model.getLongitude() +
getString(R.string.signature));

        try {
            startActivity(Intent.createChooser(i,
getString(R.string.share_location_text)));
        } catch (ActivityNotFoundException ex) {
            Toast.makeText(context, R.string.share_error,
Toast.LENGTH_SHORT).show();
        }
        Toast.makeText(MainActivity.this, R.string.share_ok,
Toast.LENGTH_SHORT).show();
    }
}

```

Listado 9: método shareLocation()

configureActivity()

Método que recoge todos aquellos elementos que tienen que inicializarse y referenciarse al cargar el activity como las variables, los layouts, el actionBar y cargar el mapFragment en su layout correspondiente.

```

private void configureActivity() {
    linearLayout = findViewById(R.id.llExit);
    linearLayout.setOnClickListener(this);
    drawerLayout = findViewById(R.id.dl);
    toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setDisplayShowTitleEnabled(false);

    actionBarDrawerToggle = new ActionBarDrawerToggle(this,
drawerLayout, toolbar, R.string.open, R.string.close);
        actionBarDrawerToggle.setDrawerIndicatorEnabled(true);

    drawerLayout.addDrawerListener(actionBarDrawerToggle);
    actionBarDrawerToggle.syncState();

    mapFragment = new MapFragment();
    FragmentManager manager = getSupportFragmentManager();
    manager.beginTransaction().replace(R.id.contendor,
mapFragment).commit();

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}

```

Listado 10: método configureActivity()

MapFragment

Fragmento que se encarga de cargar el mapa implementado de la API. Es la clase encargada de gestionar los permisos y los distintos métodos relacionados con el mapa.

onMapReady()

Método encargado de comprobar qué permisos tiene la aplicación a la hora de cargar el mapa.

```

public void onMapReady(GoogleMap googleMap) {
    mGoogleMap = googleMap;

    if (ActivityCompat.checkSelfPermission(operative,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, MY_REQUEST_INT);
        }
        return;
    } else {
        mGoogleMap.setMyLocationEnabled(true);
        getLastLocation();
    }
    mGoogleMap.getUiSettings().setCompassEnabled(false);
    this.model = PreferencesHelper.loadPreferences(operative);
    addSavedMarker();
}

```

Listado 11: método onMapReady()

onRequestPermissionsResult()

Método encargado de interpretar qué permiso permite el usuario a la hora de solicitárselos.

```

public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
        mGoogleMap.setMyLocationEnabled(true);
        getLastLocation();
        this.model = PreferencesHelper.loadPreferences(operative);
        addSavedMarker();
        mGoogleMap.getUiSettings().setCompassEnabled(false);
    } else {
        Toast.makeText(getActivity(), R.string.toast_trash,
Toast.LENGTH_SHORT).show();
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                System.exit(0);
            }
        }, TIME_WAIT);
    }
}

```

Listado 12: método onRequestPermissionsResult()

fetchLastLocation()

Método encargado de recuperar y almacenar la posición actual en la que nos encontramos.

```
private void fetchLastLocation() {
    FusedLocationProviderClient fusedLocationProviderClient =
    LocationServices.getFusedLocationProviderClient(operative);
    Task<Location> task =
    fusedLocationProviderClient.getLastLocation();
    task.addOnSuccessListener(new OnSuccessListener<Location>() {
        @Override
        public void onSuccess(Location location) {
            if (location != null) {
                currentLocation = location;
                model.setLatitude(currentLocation.getLatitude());
                model.setLongitude(currentLocation.getLongitude());
            }
        }
    });
}
```

Listado 13: método fetchLastLocation()

addNotes()

Método que interviene en la creación y solicitud de información al usuario a la hora de crear la marca en el mapa de la posición en la que ha aparcado.

```
public void addNotes() {
    fetchLastLocation();
    final AlertDialog.Builder alertDialog = new
    AlertDialog.Builder(operative, R.style.AlertDialogStyle);
    alertDialog.setTitle(R.string.dialog_notes_title);
    alertDialog.setMessage(R.string.dialog_notes_msg);

    LinearLayout layout = new LinearLayout(operative);
    LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
    lp.setMargins(40, 0, 40, 0);

    final EditText text = new EditText(operative);
    text.setTextAppearance(operative, R.style.EditText);

    layout.addView(text, lp);
    alertDialog.setView(layout);

    alertDialog.setPositiveButton(R.string.dialog_notes_save, new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            String msg = text.getText().toString();
            model.setNotes(msg);
        }
    });
}
```

```

        if (model.getNotes().isEmpty()) {
            model.setNotes(getString(R.string.def_note));
        }

        addMarkerMap();
        operative.drawerLayout.closeDrawers();
    }
});

    alertDialog.setNegativeButton(R.string.cancel, new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });

    AlertDialog alert = alertDialog.create();
    alert.show();

    Button nbutton =
alert.getButton(DialogInterface.BUTTON_NEGATIVE);

nbutton.setTextColor(getResources().getColor(R.color.orangeYellowDark));
}

```

Listado 14: método addNotes()

bitmapDescriptorFromVector()

Método encargado de transformar un `vector_asset` en un `Bitmap` utilizable por la aplicación.

```

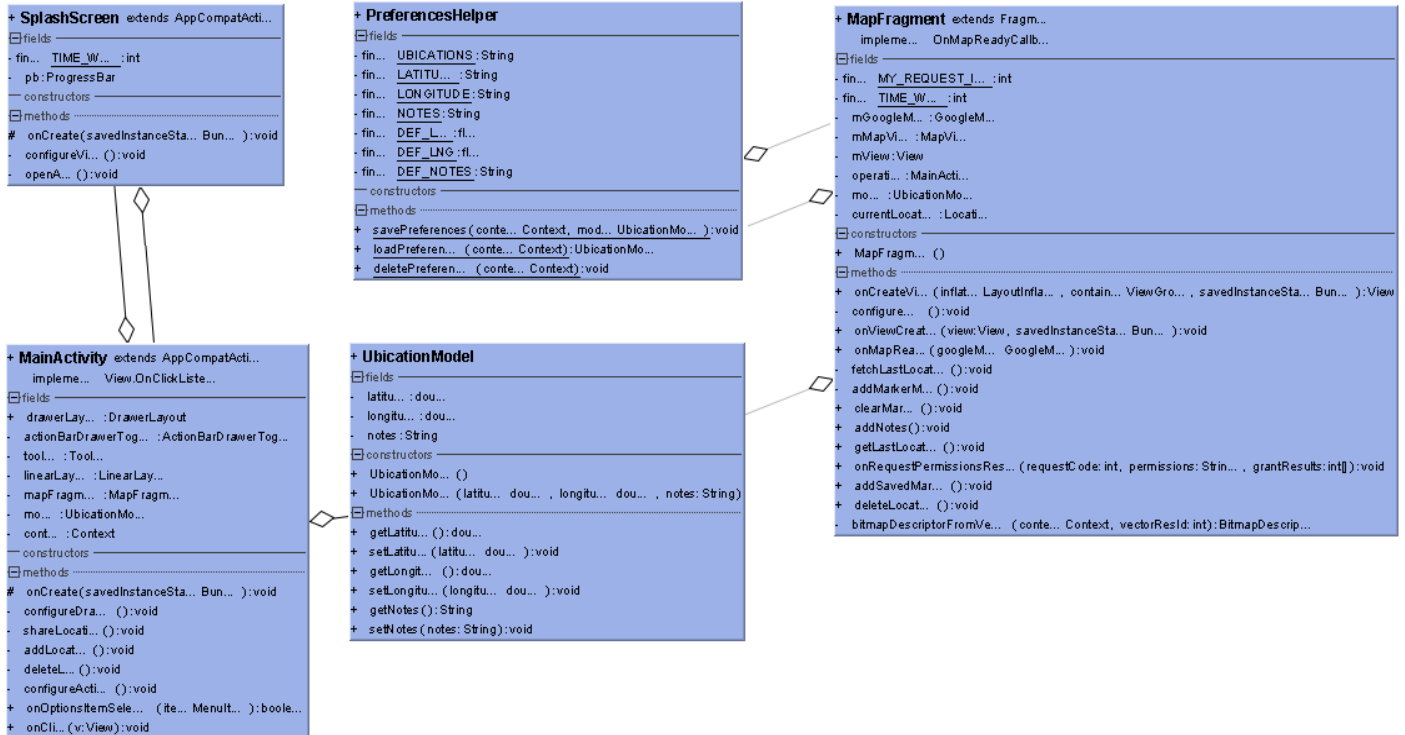
private BitmapDescriptor bitmapDescriptorFromVector(Context context, int
vectorResId){
    Drawable vectorDrawable = ContextCompat.getDrawable(context,
vectorResId);
    vectorDrawable.setBounds(0, 0,
vectorDrawable.getIntrinsicWidth(),
vectorDrawable.getIntrinsicHeight());
    Bitmap bitmap =
Bitmap.createBitmap(vectorDrawable.getIntrinsicWidth(),
vectorDrawable.getIntrinsicHeight(), Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(bitmap);
    vectorDrawable.draw(canvas);
    return BitmapDescriptorFactory.fromBitmap(bitmap);
}
}

```

Listado 15: método bitmapDescriptorFromVector()

RESULTADOS Y VALIDACIÓN

Diagrama de clases de la aplicación




Consumo de recursos

Terminales de prueba

Los terminales con los que se harán las pruebas de ejecución serán los siguientes:


Xiaomi Mi A2 Lite

CPU-Z	
SOC	DEVICE
 Qualcomm Snapdragon 625 2,02 GHz	
Cores	8
Architecture	8x ARM Cortex-A53 @ 2,02 GHz
Revision	r0p4
Process	14 nm
Clock Speed	652 MHz - 2,02 GHz
CPU 0	652 MHz
CPU 1	652 MHz
CPU 2	652 MHz
CPU 3	652 MHz
CPU 4	652 MHz
CPU 5	652 MHz
CPU 6	652 MHz
CPU 7	652 MHz
GPU Vendor	Qualcomm
GPU Renderer	Adreno (TM) 506
GPU Load	0 %
Scaling Governor	interactive

CPU-Z	
SOC	DEVICE
Model	Xiaomi Mi A2 Lite (Redmi 6 Pro) (Mi A2 Lite)
Brand	xiaomi
Board	msm8953
Hardware	qcom
Screen Size	5,84 inches
Screen Resolution	1080 x 2280 pixels
Screen Density	431 dpi
Dimensions	149,3 x 71,7 x 8,8 mm
Weight	178 g
Total RAM	3578 MB
Available RAM	1379 MB (38%)
Internal Storage	49,97 GB
Available Storage	27,23 GB (54%)
Release Date	2018-07-01

CPU-Z	
SOC	DEVICE
Android Version	9
API Level	28
Security Patch Level	2020-03-01
Bootloader	MSM8953_DAI5Y2.0_20200327 000539
Build ID	PKQ1.180917.001.V10.0.20.0.PDLMIXM
Java VM	ART 2.1.0
OpenGL ES	3.2
Kernel Architecture	aarch64
Kernel Version	3.18.120-perf-gfca2d6e (V10.0.20.0.PDLMIXM)
Root Access	No
Google Play Service	20.15.16 (100408-309763488)
System Uptime	16:47:43


Moto G 5 Plus

CPU-Z	
SOC	DEVICE
 Qualcomm Snapdragon 625 2,02 GHz	
Cores	8
Architecture	8x ARM Cortex-A53 @ 2,02 GHz
Revision	r0p4
Process	14 nm
Clock Speed	652 MHz - 2,02 GHz
CPU 0	652 MHz
CPU 1	652 MHz
CPU 2	652 MHz
CPU 3	652 MHz
CPU 4	652 MHz
CPU 5	652 MHz
CPU 6	652 MHz
CPU 7	652 MHz
GPU Vendor	Qualcomm

CPU-Z	
SOC	DEVICE
Model	Moto G (5) Plus (potter_n)
Manufacturer	motorola
Board	msm8953
Hardware	qcom
Screen Size	5,46 inches
Screen Resolution	1080 x 1920 pixels
Screen Density	403 dpi
Total RAM	2857 MB
Available RAM	1329 MB (46%)
Internal Storage	24,59 GB
Available Storage	0,89 GB (3%)

CPU-Z	
SOC	DEVICE
Android Version	8.1.0
API Level	27
Security Patch Level	2019-02-01
Bootloader	0xC092
Build ID	OPS28.85-17-6-2
Java VM	ART 2.1.0
OpenGL ES	3.2
Kernel Architecture	armv7l
Kernel Version	3.18.71-perf-g7b625a0 (77e7)
Root Access	No
Google Play Service	20.15.16 (040308-309763488)
System Uptime	00:11:45

BQ Aquaris X Pro

CPU-Z	
SOC	DEVICE
 Qualcomm Snapdragon 626 2,21 GHz	
Cores	8
Architecture	8x ARM Cortex-A53 @ 2,21 GHz
Revision	r0p4
Process	14 nm
Clock Speed	652 MHz - 2,21 GHz
CPU 0	652 MHz
CPU 1	652 MHz
CPU 2	652 MHz
CPU 3	652 MHz
CPU 4	652 MHz
CPU 5	652 MHz
CPU 6	652 MHz
CPU 7	652 MHz
GPU Vendor	Qualcomm
GPU Renderer	Adreno (TM) 506 @ 650 MHz
GPU Clock Speed	320 MHz

CPU-Z	
SOC	DEVICE
Model	Aquaris X Pro (bardock-pro)
Manufacturer	bq
Board	msm8953
Hardware	qcom
Screen Size	5,20 inches
Screen Resolution	1080 x 1920 pixels
Screen Density	423 dpi
Total RAM	3556 MB
Available RAM	1537 MB (43%)
Internal Storage	52,77 GB
Available Storage	17,05 GB (32%)

CPU-Z	
SOC	DEVICE
Android Version	8.1.0
API Level	27
Security Patch Level	2019-10-01
Bootloader	BOOT.BF.3.3
Build ID	2.11.0_20191121-1510
Java VM	ART 2.1.0
OpenGL ES	3.2
Kernel Architecture	aarch64
Kernel Version	3.18.71-perf-g4870138 (2395)
Root Access	No
Google Play Service	20.15.16 (040408-309763488)
System Uptime	1 days, 18:07:39

Redmi note 8 Pro

CPU-Z	
SOC	DEVICE
 ARM Cortex-A55 ARM Cortex-A76 2,05 GHz	
Cores	8
big.LITTLE	Yes (2 clusters)
Architecture	2x ARM Cortex-A76 @ 2,05 GHz 6x ARM Cortex-A55 @ 2,00 GHz
Clock Speed	774 MHz - 2,05 GHz
CPU 0	2000 MHz
CPU 1	2000 MHz
CPU 2	2000 MHz
CPU 3	2000 MHz
CPU 4	2000 MHz
CPU 5	2000 MHz
CPU 6	1733 MHz
CPU 7	1733 MHz
GPU Vendor	ARM
GPU Renderer	Mali-G76 MC4
GPU Load	0 %
Scaling Governor	unknown

CPU-Z	
SOC	DEVICE
Model	Redmi Note 8 Pro (begonia_eea)
Manufacturer	Xiaomi
Brand	Redmi
Board	begonia
Hardware	mt6785
Screen Size	6,53 inches
Screen Resolution	1080 x 2340 pixels
Screen Density	394 dpi
Total RAM	5676 MB
Available RAM	1882 MB (33%)
Internal Storage	110,05 GB
Available Storage	79,43 GB (72%)

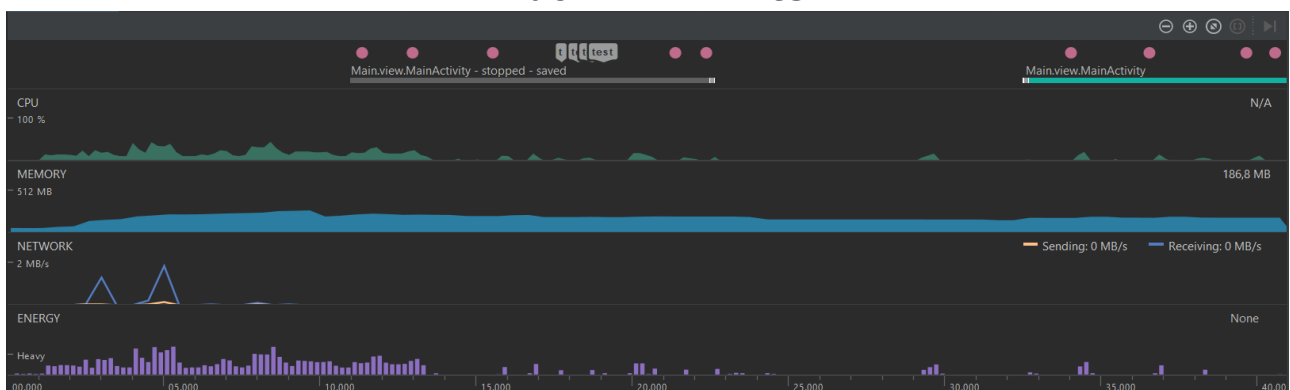
CPU-Z	
SOC	DEVICE
Android Version	9
API Level	28
Security Patch Level	2019-11-01
Bootloader	unknown
Build ID	PPR1.180610.011
Java VM	ART 2.1.0
OpenGL ES	3.2
Kernel Architecture	aarch64
Kernel Version	4.14.94+ (V11.0.3.0.PGGEUXM)
Root Access	No
Google Play Service	20.15.16 (100400-309763488)
System Uptime	18 days, 19:13:53

Se registran pruebas de ejecución para efectuarlas con los dispositivos, siendo la lista de tareas las siguientes:

1. Abrir app dar permisos
2. Desplazar mapa
3. Botón de centrar mapa
4. Abrir NavigationDrawer
5. Guardar localización → Alert Cancelar
6. Guardar localización → Test → Guardar
7. Compartir aparcamiento
8. Borrar aparcamiento → Alert Cancelar
9. Borrar aparcamiento → Alert Aceptar
10. Salir

Para lo cual los resultados con los diferentes terminales han sido

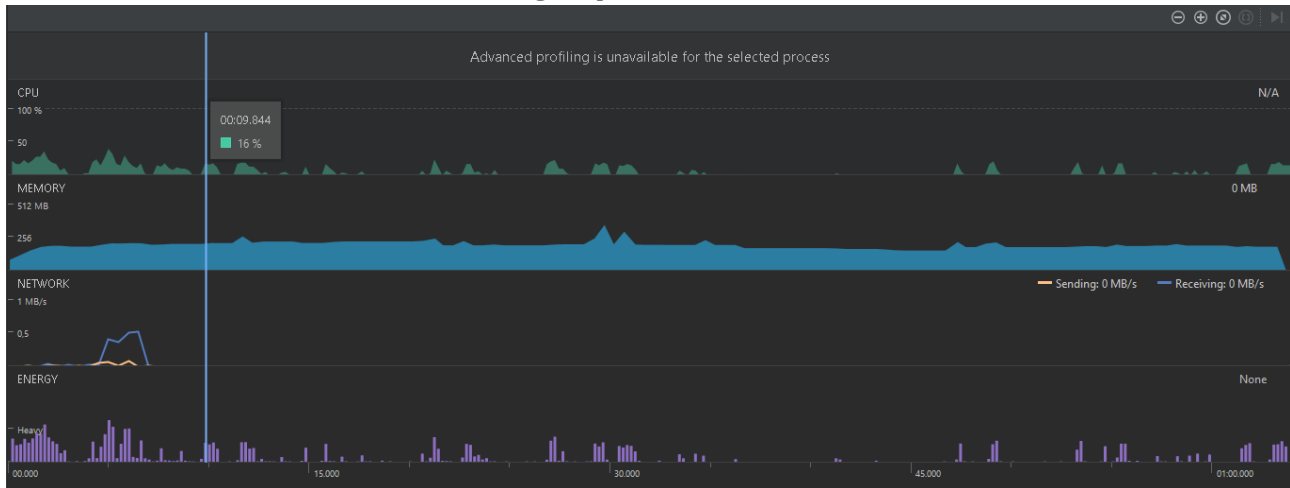
Xiaomi Mi A2 Lite



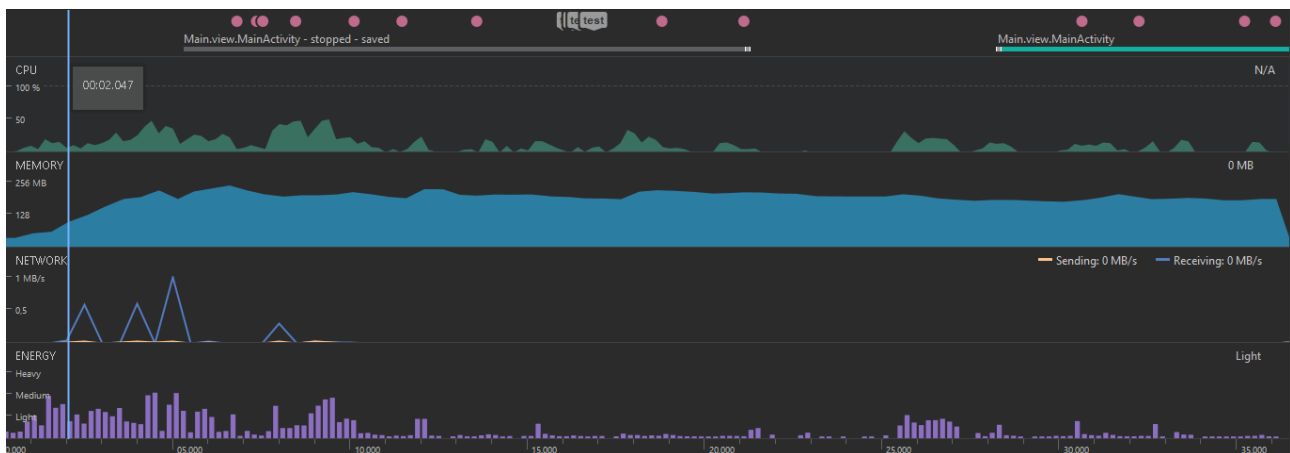
Moto G 5 Plus



BQ Aquaris X Pro



Redmi Note 8 Pro



Como se puede observar, el rendimiento de la aplicación es muy similar en todos los casos ya que si desglosamos los pasos de la ejecución de pruebas se observa que:

Recursos generales

Abrir la app totalmente requiere un tiempo aproximado de 10 segundos comenzando a contar desde el punto de la instalación. Durante este tiempo se ejecuta toda la funcionalidad de la aplicación, desde cargar el Splash, ubicarnos en el mapa con la API de Google Maps, hasta que no está completamente funcional, no se ejecuta el main activity. Para que todo esto tenga lugar se pueden observar diferentes funcionamientos dentro del terminal.

Consumo de CPU

La CPU tiene su pico de funcionamiento en la aplicación en el momento de cargar toda la app, siendo su punto más elevado en un 50% del rendimiento máximo del componente, en ningún caso se está saturando el dispositivo.

Teniendo en cuenta que las pruebas están hechas sobre procesadores de 2016 (cuatro años de antigüedad a fecha de este proyecto) se considera que el funcionamiento en lo que refiere a potencia requerida de procesador está más que comedido teniendo en cuenta la todo el proceso previo en el funcionamiento de la app.

Consumo de RAM

En la RAM del mismo modo que en el apartado de la CPU se observa su pico máximo de funcionamiento en el momento de carga completa de la aplicación, llegándose a consumir un máximo de 0,5GB, que paulatinamente va reduciéndose hasta quedar aproximadamente en unos 100MB.

Desde el comienzo se planeó la máxima velocidad posible en el uso, enfocándose todo a una experiencia de usuario rápida y limpia, por ello el consumo de este recurso se ha hecho deliberadamente alto desde el punto de vista de la programación del código, con el fin de que prevalezca la velocidad de la aplicación.

Consumo de datos

En este apartado se observa como en los anteriores, que en el arranque de la aplicación se llegan a picos de velocidad más que asumibles en las conexiones de datos que manejamos en la actualidad en las redes móviles, ya que este pico máximo es de unos 2MB/s haciéndose una carga de 2MB.

Estos requisitos de red tan bajos hacen que para que la aplicación tenga la usabilidad que se busca, sea suficiente con tener disponible una conexión de datos de al menos del tipo 3G.

Para ello mostramos un mapa de cobertura 3G a nivel nacional en el que podemos comprobar que es un tipo de red ampliamente extendida.



No obstante, de disponer de un tipo de red más lenta, no se impide el uso de la aplicación, simplemente el uso no será tan rápido ya que la descarga de esos 2MB de arranque aproximados.

En caso de disponer la cobertura de datos más lenta (GPRS) nos proporciona aproximadamente 80Kbps, lo cual hace que el arranque inicial de datos se diera en 25 segundos (sería el peor de los escenarios), por lo que se hace un tiempo aceptable teniendo en cuenta la velocidad media de uso de las aplicaciones.

Consumo de energía

En el apartado energético comprobamos que como es común, el consumo más elevado se da en el momento de instalar la aplicación, ya que se da el aporte energético más alto al procesador y a la memoria RAM. Este tipo de consumo siempre va a ser variable dependiendo del tipo de procesador (nanómetros de la litografía) y de la tecnología de memoria RAM (LPDDR1, LPDDR2, LPDDR3...).

Mención especial en el apartado energético hay que hacer en cuanto al funcionamiento de la geolocalización en tiempo real de la API de Google Maps utilizada, ya que esta busca cambios de la ubicación aproximadamente con latencias de 1 segundo, lo que provoca picos de consumo en la batería para alimentar a la RAM y a la CPU.

CONCLUSIONES

En *Where I Park* se han implementado soluciones en lo que refiere a la velocidad de ubicación basada en la API que proporciona Google Maps, minimizando el uso de datos, así como enfocándose en una ejecución del código lo más limpia y efectiva posible.

Destacamos mediante una planificación del trabajo basada en la entrega de sprints diarios, la obtención de los diferentes hitos propuestos en plazos de tiempo mucho más cortos que los inicialmente previstos, consiguiéndose con esta velocidad de trabajo ampliar las expectativas de la aplicación hasta el punto de desarrollar varias versiones con funcionalidades adicionales a las proyectadas en un primer momento.

Debido a los continuos avances en el desarrollo de aplicaciones para sistemas operativos móviles, es imprescindible mantener en una línea de tiempo prolongada una rama de pruebas en la que buscar un continuo aprendizaje y mejora del código previo.

Innovación

Tal y como se enfatiza durante todo el documento, *Where I Park* ofrece una solución rápida, concreta y eficaz.

Sin duda, es una tendencia solucionar problemas concretos y alejarse de aplicaciones con funcionalidades masivas que aportan más opciones de las que el usuario realmente demanda.

Trabajo futuro

Basándonos en la experiencia de uso y aceptación de *Where I Park*, se piensa desarrollar nuevas “suites” de pequeñas funcionalidades a demanda de los usuarios.

BIBLIOGRAFÍA Y WEBGRAFÍA

Google Maps API

- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 14, 2020 de <https://developer.android.com/guide/topics/security/permissions?hl=es-419>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 14, 2020 de <https://developer.android.com/training/permissions/requesting?hl=es-419>
- HERMOSA PROGRAMACIÓN. (2020) <http://www.hermosaprogramacion.com/> Fecha de consulta: abril 14, 2020 de <http://www.hermosaprogramacion.com/2016/05/google-maps-android-api-v2/>
- GOOGLE DEVELOPERS. (2020) <https://developers.google.com/> Fecha de consulta: abril 14, 2020 de <https://developers.google.com/maps/documentation/javascript/controls>
- SGOLIVER. (2016) <https://www.sgoliver.net/> Fecha de consulta: abril 14, 2020 de <https://www.sgoliver.net/blog/mapas-en-android-google-maps-android-api-1/>
- GOOGLE DEVELOPERS. (2020) <https://developers.google.com/> Fecha de consulta: abril 14, 2020 de https://developers.google.com/maps/documentation/android-sdk/location#code_samples
- JAVATIPS. (2019) <https://www.javatips.net/> Fecha de consulta: abril 14, 2020 de <https://www.javatips.net/api/android-samples-master/ApiDemos/app/src/main/java/com/example/mapdemo/PermissionUtils.java>
- GOOGLE DEVELOPERS. (2020) <https://developers.google.com/> Fecha de consulta: abril 14, 2020 de <https://developers.google.com/maps/documentation/android-sdk/location>
- GOOGLE DEVELOPERS. (2019) <https://developers.google.com/> Fecha de consulta: abril 14, 2020 de <https://developers.google.com/android/reference/com/google/android/gms/maps/UiSettings>

- STACKOVERFLOW (2014) <https://stackoverflow.com/>. Fecha de consulta: abril 14, 2020 de <https://stackoverflow.com/questions/14376361/disable-center-button-in-mylocation-at-google-map-api-v2>
- MEDIUM. (2017) <https://medium.com/>. Fecha de consulta: abril 15, 2020 de <https://medium.com/@victor.garibayy/obteniendo-mi-ubicaci%C3%B3n-en-android-studio-377226910823>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 15, 2020 de <https://developer.android.com/reference/android/support/v4/app/ActivityCompat.OnRequestPermissionsResultCallback>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 15, 2020 de <https://developer.android.com/reference/kotlin/android/location/LocationManager>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: abril 15, 2020 de <https://stackoverflow.com/questions/36659038/get-current-location-in-onmapready-in-android-using-google-locations-api>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 15, 2020 de <https://stackoverflow.com/questions/53257995/how-to-create-fusedlocationproviderclient-in-a-service-that-can-run-until-stop>
- STACKOVERFLOW (2019) <https://stackoverflow.com/>. Fecha de consulta: abril 15, 2020 de <https://stackoverflow.com/questions/55396046/fusedlocationproviderclient-not-returning-location>
- STACKOVERFLOW (2019) <https://stackoverflow.com/>. Fecha de consulta: abril 15, 2020 de <https://stackoverflow.com/questions/54352761/why-is-this-code-which-uses-fusedlocationproviderclient-to-get-the-users-locat>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 15, 2020 de <https://stackoverflow.com/questions/49341694/i-used-fusedlocationproviderclient-to-get-location-from-user-so-i-just-want-whe>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 16, 2020 de <https://developer.android.com/training/location/retrieve-current?hl=es-419>
- GOOGLE DEVELOPERS. (2019) <https://developers.google.com/>. Fecha de consulta: abril 16, 2020 de

- <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 16, 2020 de <https://stackoverflow.com/questions/48206319/check-permission-for-user-location>
 - STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 16, 2020 de <https://stackoverflow.com/questions/48794204/missing-permissions-required-by-fusedlocationclient>
 - STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 16, 2020 de <https://stackoverflow.com/questions/51800657/permissionutils-can-not-be-resolved>
 - STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: abril 16, 2020 de <https://stackoverflow.com/questions/35521756/error-package-permissionutils-does-not-exist>
 - ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 16, 2020 de <https://developer.android.com/training/permissions/requesting?hl=es-419>
 - STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 17, 2020 de <https://stackoverflow.com/questions/13932441/android-google-maps-v2-set-zoom-level-for-mylocation>
 - STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 17, 2020 de <https://stackoverflow.com/questions/29868121/how-do-i-zoom-in-automatically-to-the-current-location-in-google-maps-api-for-an>
 - ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 17, 2020 de <https://developer.android.com/training/location?hl=es-419>
 - ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/training/location/retrieve-current>
 - ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/jetpack/androidx/migrate>
 - GOOGLE DEVELOPERS. (2020) <https://developers.google.com/>. Fecha de consulta: abril 20, 2020 de <https://developers.google.com/admin-sdk/directory/v1/reference/>

- GOOGLE DEVELOPERS. (2020) <https://developers.google.com/>. Fecha de consulta: abril 20, 2020 de <https://developers.google.com/apis-explorer>
- GOOGLE DEVELOPERS. (2017) <https://developers.google.com/>. Fecha de consulta: abril 21, 2020 de <https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap.OnMyLocationButtonClickListener?hl=es-419>
- GOOGLE DEVELOPERS. (2020) <https://developers.google.com/>. Fecha de consulta: abril 21, 2020 de <https://developers.google.com/maps/documentation?hl=es>
- GOOGLE DEVELOPERS. (2020) <https://developers.google.com/>. Fecha de consulta: abril 20, 2020 de <https://developers.google.com/maps/documentation/android-sdk/v3-client-migration?hl=es>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: abril 30, 2020 de <https://stackoverflow.com/questions/45595004/how-to-disable-google-maps-realign-north-button>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: mayo 7, 2020 de <https://stackoverflow.com/questions/42365658/custom-marker-in-google-maps-in-android-with-vector-asset-icon/45564994>

Comunicación Activity/Fragment

- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/reference/android/support/v4/app/FragmentManager.html>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/reference/kotlin/androidx/appcompat/app/ActionBarDrawerToggle>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/reference/android/app/FragmentManager>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/reference/android/app/ActivityGroup>

- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 19, 2020 de <https://developer.android.com/reference/android/widget/ZoomButton>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 21, 2020 de <https://developer.android.com/training/basics/fragments/communicating>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 21, 2020 de <https://developer.android.com/reference/android/app/Activity.html#SavingPersistentState>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: abril 21, 2020 de <https://developer.android.com/training/basics/fragments/communicating?hl=es>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/59776719/android-pass-instance-of-mainactivity-to-fragment>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/37653037/android-how-to-call-fragment-from-my-main-activity>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/46150954/call-method-from-fragment-to-activity>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/35747346/how-to-call-fragment-method-from-main-activity>

Navigation Drawer + Map Fragment

- STACKOVERFLOW (2015) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/31875981/run-map-fragment-from-navigation-drawer>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/43707970/android-error-with-map-fragment-on-navigation-drawer-and-permission-check>

- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/47937252/map-fragment-on-navigation-drawer-cant-access-current-location>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: abril 21, 2020 de <https://stackoverflow.com/questions/35825218/mapfragment-in-android-navigation-drawer-causes-android-view-inflateexception>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 2, 2020 de <https://developer.android.com/guide/topics/resources/runtime-changes?hl=es-419>
- STACKOVERFLOW (2015) <https://stackoverflow.com/>. Fecha de consulta: mayo 3, 2020 de <https://stackoverflow.com/questions/30543605/how-to-add-footer-to-navigationview-android-support-design-library/30611068#30611068>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: mayo 3, 2020 de <https://es.stackoverflow.com/questions/100573/navigationview-appmenu-cambiar-tama%C3%B1o-de-letra>
- STACKOVERFLOW (2015) <https://stackoverflow.com/>. Fecha de consulta: mayo 3, 2020 de <https://stackoverflow.com/questions/33407448/change-color-of-navigation-drawer-icon-in-android-studio-default-template>
- <https://stackoverflow.com/questions/19194526/how-to-close-navigation-drawer-when-an-item-is-pressed-from-it>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 4, 2020 de <https://developer.android.com/reference/android/app/AlertDialog>
- MEDIUM (2019) <https://medium.com/>. Fecha de consulta: mayo 4, 2020 de <https://medium.com/@suragch/creating-a-custom-alertdialog-bae919d2efa5>
- STACKOVERFLOW (2014) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/21123653/how-to-center-text-in-alert-dialog-builder/22595965>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/39930505/change-color-of-button-in-alertdialog-builder>

- STACKOVERFLOW (2013) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/16405614/modifying-edittext-color-in-alertdialog-wont-work>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 4, 2020 de <https://developer.android.com/guide/topics/ui/dialogs?hl=es-419>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/47663789/change-the-color-of-the-input-line-in-the-alert-dialog>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/5350607/resize-edit-box-inside-alert-dialog-in-android>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/40480637/change-size-of-edittextany-view-shown-in-alertbox>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: mayo 4, 2020 de <https://stackoverflow.com/questions/9345735/resizing-edittext-inside-of-an-alertdialog>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: mayo 5, 2020 de <https://stackoverflow.com/questions/46397707/how-to-change-color-of-button-in-alert-dialog>

Toolbar personalizada

- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 30, 2020 de <https://developer.android.com/reference/android/widget/Toolbar>
- STACKOVERFLOW (2017) <https://stackoverflow.com/>. Fecha de consulta: abril 30, 2020 de <https://stackoverflow.com/questions/46066434/android-navigationdrawer-without-actionbar/46066548>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: abril 30, 2020 de <https://es.stackoverflow.com/questions/162173/como-modificar-el-actionbar-de-navigation-drawer-activity>

- MEDIUM (2016) <https://medium.com/@ssaurel/implement-a-navigation-drawer-with-a-toolbar-on-android-m-68162f13d220>. Fecha de consulta: abril 30, 2020 de
- STACKOVERFLOW (2017) <https://stackoverflow.com/questions/28327081/transparent-actionbar-and-statusbar-in-android-lollipop>. Fecha de consulta: abril 30, 2020 de
- STACKOVERFLOW (2016) <https://stackoverflow.com/questions/37138485/how-to-set-toolbar-instead-of-actionbar>. Fecha de consulta: abril 30, 2020 de
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/training/appbar/setting-up>. Fecha de consulta: abril 30, 2020 de

Launcher / Splash Screen

- ANDROID DEVELOPERS. (2020) <https://developer.android.com/reference/android/widget/ProgressBar>. Fecha de consulta: abril 27, 2020 de
- MEDIUM (2017) <https://medium.com/@jelse/the-complete-android-splash-screen-guide-c7db82bce565>. Fecha de consulta: abril 29, 2020 de
- STACKOVERRUN (2012) <https://stackoverflow.com/es/q/3432354>. Fecha de consulta: abril 29, 2020 de
- LOGICCHIP (2020) <https://www.logicchip.com/android-splash-screen-progress-bar/>. Fecha de consulta: abril 29, 2020 de
- STACKOVERFLOW (2017) <https://stackoverflow.com/questions/42017161/android-progress-bar-on-splash-screen>. Fecha de consulta: abril 29, 2020 de
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/reference/android/graphics/drawable/LayerDrawable>. Fecha de consulta: abril 29, 2020 de

- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: abril 29, 2020 de <https://stackoverflow.com/questions/38161959/android-text-in-drawable-layer-list>

Shared Preferences

- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/reference/android/content/SharedPreferences.Editor>
- STACKOVERFLOW (2011) <https://stackoverflow.com/>. Fecha de consulta: mayo 1, 2020 de <https://stackoverflow.com/questions/5950043/how-to-use-getsharedpreferences-in-android>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/studio/profile/monitor?hl=es>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: mayo 1, 2020 de <https://stackoverflow.com/questions/52687504/how-to-open-android-device-monitor-in-latest-android-studio-3-2>
- STACKOVERFLOW (2016) <https://stackoverflow.com/>. Fecha de consulta: mayo 1, 2020 de <https://stackoverflow.com/questions/34603355/android-device-monitor-data-folder-is-empty>
- STACKOVERFLOW (2012) <https://stackoverflow.com/>. Fecha de consulta: mayo 1, 2020 de <https://stackoverflow.com/questions/11741270/android-sharedpreferences-in-fragment>
- STACKOVERFLOW (2012) <https://stackoverflow.com/>. Fecha de consulta: mayo 1, 2020 de <https://stackoverflow.com/questions/12612445/android-load-shared-preferences-in-main-activity>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/training/data-storage/shared-preferences?hl=es#java>
- STACKOVERFLOW (2015) <https://stackoverflow.com/>. Fecha de consulta: mayo 1, 2020 de <https://stackoverflow.com/questions/32333758/android-delete-sharedpreferences>

Comunicación entre apps

- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/training/sharing/receive?hl=es>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/training/sharing/send?hl=es>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/guide/components/intents-filters?hl=es-419>
- ANDROID DEVELOPERS. (2019) <https://developer.android.com/>. Fecha de consulta: mayo 1, 2020 de <https://developer.android.com/training/location/display-address?hl=es>
- ANDROFAST. (2018) <https://www.androfast.com/> Fecha de consulta: mayo 1, 2020 de <https://www.androfast.com/2017/09/como-guardar-la-direccion-y-coordenadas.html>
- STACKOVERFLOW (2018) <https://stackoverflow.com/>. Fecha de consulta: mayo 2, 2020 de <https://stackoverflow.com/questions/50601295/how-can-you-share-location-same-like-whatsapp-in-android-with-chooser-dialog>
- STACKOVERFLOW. (2014) <https://stackoverflow.com/>. Fecha de consulta: mayo 2, 2020 de <https://stackoverflow.com/questions/23573444/android-development-how-can-i-share-the-location-of-a-marker-i-place-on-google>
- SENDBIRD. (2020) <https://blog.sendbird.com/>. Fecha de consulta: mayo 2, 2020 de <https://blog.sendbird.com/map-preview-and-location-sharing-on-android/>

JavaDoc

- STACKOVERFLOW. (2015) <https://stackoverflow.com/>. Fecha de consulta: mayo 8, 2020 de <https://stackoverflow.com/questions/29162820/how-to-create-javadoc-using-android-studio-without-r-and-buildconfig>

- COROCHANNNOTE. (2015) <https://corochann.com/>. Fecha de consulta: mayo 8, 2020 de <https://corochann.com/javadoc-coding-rule-of-link-linkplain-see-372.html>
- APRENDERAPROGRAMAR. (2017) <https://www.aprenderaprogramar.com/>. Fecha de consulta: mayo 8, 2020 de https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:ique-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163

Documentación para la memoria

- TIOBE. (2020) <https://www.tiobe.com/> Fecha de consulta: abril 27, 2020 de <https://www.tiobe.com/tiobe-index/>
- ANDROID. (2020) <https://www.android.com/>. Fecha de consulta: abril 27, 2020 de https://www.android.com/intl/es_es/history/#/marshmallow
- XATAKA. (2018) <https://www.xatakandroid.com/>. Fecha de consulta: abril 27, 2020 de <https://www.xatakandroid.com/sistema-operativo/historia-y-evolucion-de-android-como-un-sistema-operativo-para-cameras-digitales-acabo-conquistando-los-moviles>
- ANDROID DEVELOPERS. (2020) <https://developer.android.com/>. Fecha de consulta: abril 27, 2020 de <https://developer.android.com/about/dashboards?hl=es-419>
- WIKIPEDIA. (2020) <https://es.wikipedia.org/wiki/Wikipedia:Portada>. Fecha de consulta: mayo 8, 2020 de [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))
- KANBANIZE. (2020) <https://kanbanize.com/>. Fecha de consulta: mayo 8, 2020 de <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
- WIKIPEDIA. (2020) <https://es.wikipedia.org/wiki/Wikipedia:Portada>. Fecha de consulta: mayo 8, 2020 de https://es.wikipedia.org/wiki/Desarrollo_de_software
- WIKIPEDIA. (2020) <https://es.wikipedia.org/wiki/Wikipedia:Portada>. Fecha de consulta: mayo 11, 2020 de [https://en.wikipedia.org/wiki/List_of_Qualcomm_Snapdragon_systems-on-chip#Snapdragon_617,_625_and_626_\(2015/16\)](https://en.wikipedia.org/wiki/List_of_Qualcomm_Snapdragon_systems-on-chip#Snapdragon_617,_625_and_626_(2015/16))

- PROANDROID. (2018) <https://www.proandroid.com/>. Fecha de consulta: mayo 12, 2020 de <https://www.proandroid.com/tipos-red-movil-existentes/>
- MOVISTAR. (2020) <http://www.movistar.es/>. Fecha de consulta: mayo 12, 2020 de <http://www.movistar.es/particulares/coberturas/movil/4G/>