# **ESTUDIO PREVIO PARA TFG**

## ***DISPOSITIVOS***

### ¿Qué sistema operativo elegir?

Para elegir el sistema operativo de los dispositivos en los que va a ser funcional la app, se va a analizar el número de usuarios que utiliza cada sistema operativo. A pesar de que en 2010 la cuota de mercado de smartphones por sistema operativo se repartía de una forma más equitativa, en la actualidad hay dos grandes sistemas operativos que copan la cuota de mercado: iOS y Android.

Según los datos de la consultora de tecnología IDC [1], que ha analizado el uso de los distintos sistemas operativos en dispositivos móviles, la cuota de mercado de dispositivos con el sistema operativo iOS en el año 2021 es de un 16’2%, con una previsión de bajada que le llevaría al 15’1% en el año 2025.

Mientras tanto, la cuota de mercado de dispositivos con el sistema operativo Android en el año 2021 es de un 83’8%, teniendo una previsión de subida que, según el análisis de IDC, llevaría al sistema operativo de Google hasta una cuota de mercado del 84’9% en el año 2025.

De este mismo estudio sacamos el dato de que la cuota de mercado de otros sistemas operativos en dispositivos móviles se mantendrá sobre el 0% a lo largo de los próximos años, debido a este duopolio de iOS y Android.

¿ELABORAR GRÁFICO CON PORCENTAJES?

Por tanto, tras analizar este estudio de la consultora IDC, podemos deducir que la mejor opción es desarrollar la app para el sistema operativo Android, al ser el sistema operativo que tiene mayor cuota de mercado entre los dispositivos móviles, con una previsión de subida a medio plazo.

Además de analizar la viabilidad del sistema operativo desde el punto de vista del posible cliente final, se va a analizar la viabilidad del sistema operativo desde el punto de vista del desarrollador de la app. Para la publicación final de la app, tanto la Play Store de Google como la App Store de Apple exigen una tarifa de registro. En el caso de la Play Store, la tarifa es un pago único al crear la cuenta de desarrollador de 25$; sin embargo, en el caso de la App Store se exige una cuenta de desarrollador cuya cuota cuesta 99$ y que debe ser renovada anualmente.

La opción de desarrollar la app para Android se ve respaldada de esta forma debido a los elevados costes económicos por parte de la App Store, que dificulta el mantenimiento en el tiempo de las apps al tener unos precios tan elevados.

### Versiones del sistema operativo

Una vez se ha decidido que el sistema operativo para el que se destinará la app será Android, toca estudiar cuál será la versión mínima del sistema operativo que podrá ejecutar dicha app. Para ello, se va a analizar el uso de las distintas versiones en la actualidad, buscando una estimación de cara al año 2023, cuando podría comenzar a tener uso la app en su versión definitiva, cuando se haya lanzado el producto final.

La última versión de Android que ha sacado al mercado Google ha sido Android 12, cuya fecha de lanzamiento fue el 4 de octubre de 2021 [2]. Actualmente, Google se encuentra desarrollando la nueva versión de su sistema operativo, llamado Android 13 [3], del cual ha sacado su primera versión para desarrolladores (Developer Preview) el pasado 10 de febrero de 2022. Según el calendario de lanzamientos, estaría previsto que el lanzamiento final de Android 13 sea al final del verano, entre los meses de agosto y septiembre.

A pesar de disponer de fechas cerradas de lanzamientos, como es el caso de Android 12, los dispositivos móviles no tienen actualizaciones a estas nuevas versiones desde la fecha exacta del lanzamiento. Las actualizaciones a las nuevas versiones van saliendo poco a poco, en función del trabajo de cada fabricante de smartphones, siendo los dispositivos Google Pixel los primeros en recibir estas actualizaciones.

Conociendo estos datos, podemos estimar que, de cara al año 2023, las versiones de Android 12 y Android 13 tendrán una cuota de mercado no muy grande.

Otro dato para tener en cuenta es el tiempo que tardan los usuarios en renovar su smartphone, adquiriendo uno nuevo, algo que puede servir para orientarnos a la hora de acotar la versión mínima de Android para la que se va a desarrollar nuestra app. Según un sondeo realizado por IO Investigación [4] sobre una muestra de individuos con una edad entre los 18 y los 65 años residentes en España, el intervalo favorito de los encuestados para cambiar de smartphone es entre los tres y cuatro años de uso, opción escogida por el 44’41% de los sondeados, seguida del intervalo entre el primer y segundo año de uso del smartphone, elegido por el 39’11% de los sondeados. Así, se puede considerar que, para los españoles, el momento clave para cambiar de smartphone es entre el segundo y el tercer año de uso del dispositivo.

¿ELABORAR GRÁFICO CON PORCENTAJES DE CAMBIO DE SMARTPHONE?

Además, los smartphones van recibiendo diversas actualizaciones, las cuales aplican una parte de los usuarios conforme las marcas van recibiendo dichas actualizaciones. Como se ha comentado previamente, no todas las reciben al mismo ritmo, además de que las actualizaciones van llegando de manera progresiva dentro de los distintos dispositivos que tienen en su catálogo las marcas. Por tanto, el objetivo es intentar acotar y estimar qué versión mínima de Android es la óptima para el desarrollo de nuestra app.

Tras estudiar la cuota de mercado de los dispositivos con el sistema operativo Android y la tendencia de los usuarios a la hora de cambiar de smartphone, vamos a ver qué versiones de Android son las más usadas, según datos de enero del año 2022 [5].

Si se tiene en cuenta solo el dato de los smartphones en enero del año 2022, podemos observar que Android 11 es la versión que tiene mayor cuota de mercado, con un 37’92% del total de los usuarios de Android. Le sigue de cerca Android 10, con un 26’02%, siendo la tercera versión más utilizada Android 9.0 Pie, con un 12’48%. Un dato para considerar es la poca presencia en dispositivos móviles de Android 12, la versión más reciente, que la tienen tan solo un 3’81% de los usuarios con smartphones cuyo sistema operativo es de Android, siendo 8.1 Oreo la cuarta versión más utilizada. Analizando la tendencia de las distintas versiones, tanto Android 11 como Android 12 se encuentran en pleno auge de usuarios, en detrimento de las versiones Android 8.1 Oreo, Android 9.0 Pie y Android 10.

Viendo la tendencia de los usuarios a la hora de renovar sus smartphones y la tendencia que están siguiendo las distintas versiones de Android en [5], se puede confirmar que la versión 8.1 Oreo tendrá un porcentaje de usuarios en torno al 4%, mientras que la versión 9.0 Pie será usada por menos de un 10% de los usuarios de Android, ya que es una de las versiones que mayor bajada de usuarios está teniendo junto a Android 10, a pesar de las buenas cifras de usuarios de la versión que sacó Google en septiembre del año 2019.

¿INSERTAR GRÁFICO DE LAS VERSIONES OBTENIDO DE [5]?

Respecto al uso de tablets, este dispositivo no ha sido considerado durante todo el estudio realizado para determinar la versión mínima de Android. Sin embargo, si se incluye al análisis los datos de usuarios de tablets cuyo sistema operativo sea Android se obtiene una tendencia muy similar [6] a la tendencia observada con los usuarios de smartphones, por lo que no afectaría al estudio su consideración.

¿INSERTAR GRÁFICO DE LAS VERSIONES OBTENIDO DE [6]?

Tras este análisis, considerando la evolución en el porcentaje de usuarios de cada versión de Android, teniendo en cuenta cada cuánto tiempo suelen renovar los usuarios sus smartphones, y estimando que la puesta en producción de la app se encuentre entre finales del año 2022 y principios del año 2023, se puede concluir que la versión mínima para la que se debe desarrollar la app es para Android 10.

## ***TECNOLOGÍAS PARA EL DESARROLLO***

Tras haber investigado las tendencias de los usuarios en la compra y ciclo de vida de los smartphones con vistas a elegir el sistema operativo para el que estará destinada la app y su versión mínima, el siguiente paso es decidir con qué tecnologías desarrollar la app y completar el proyecto.

Para ello, se deben elegir varias tecnologías factibles para el desarrollo, analizarlas con sus ventajas e inconvenientes y, finalmente, elegir cuáles serán las que se usen a lo largo de este proyecto. Entre las distintas tecnologías existentes, se deben elegir el lenguaje de programación, el entorno de desarrollo, o un posible gestor de base de datos, todo con vistas al correcto funcionamiento del producto final.

### Lenguaje de programación

La elección del lenguaje de programación para el desarrollo de la app es una decisión de gran importancia que debe tomarse antes de comenzar a realizar pruebas con código. Para ello, primero hay que saber qué opciones hay, analizando los posibles beneficios e inconvenientes de cada uno, para elegir finalmente en qué lenguaje se va a trabajar.

Para la elección del lenguaje de programación para este proyecto se van a valorar dos opciones: Java y Kotlin. Sin embargo, existen más lenguajes de programación que puedan ser utilizados, sobre todo si se realiza una aplicación híbrida (que funcione en distintos sistemas operativos móviles). En este caso, como la app será desarrollada para Android, no se tendrán en cuenta dichos lenguajes de programación para la elección, debido en parte al desconocimiento por parte del alumno de estos lenguajes.

Java, de Oracle, es un lenguaje de programación muy usado en todo el mundo, para todo tipo de plataformas, siendo el tercer lenguaje más popular en GitHub. De hecho, el propio sistema operativo de Android está desarrollado en Java (junto a C y C++), lo que hace que sea una de las principales opciones para los desarrolladores a la hora de escribir apps.

Kotlin, de JetBrains, es un lenguaje de programación que busca mejorar ciertos aspectos de Java, siendo apoyado por Google desde el año 2017 para el desarrollo de apps para Android. Debido a la legibilidad de su código y a su rápida curva de aprendizaje, en poco tiempo se convirtió en una de las opciones favoritas de los programadores, ya que no solo es válido para Android, también lo es para otras plataformas, incluyendo la propia iOS.

Son muchas más las ventajas que ofrece Kotlin. Un claro ejemplo se puede observar a la hora de realizar proyectos en Kotlin, en los que pueden convivir archivos programados en este lenguaje de JetBrains junto con archivos programados en Java. También es un lenguaje muy flexible, que permite que el desarrollador escriba el código de la forma que más le guste.

Sin embargo, también tiene ciertos aspectos en contra. Su principal aspecto en contra es que, al tener una comunidad de desarrolladores más pequeña que la de Java, hay menos soporte que sirva de ayuda a personas que estén comenzando con este lenguaje, teniendo menos bibliotecas, tutoriales o publicaciones de ayuda en la red. Al principio, debido a la sintaxis concisa de este lenguaje, incluso puede ser complicado realizar un cambio a este lenguaje.

Vamos a valorar también la experiencia personal del alumno para poder tomar una decisión. El alumno tiene experiencia con el lenguaje Java, que ha usado tanto en asignaturas del Grado como en diversos proyectos que ha realizado a nivel personal, para su propio uso. Respecto a Kotlin, el alumno no se ha enfrentado nunca a proyectos cuyo código estuviese escrito en este lenguaje. Por tanto, existen dos opciones: con Java, el alumno se quedaría en su “zona de confort”, programando con un lenguaje conocido; mientras, con Kotlin, el alumno se enfrentaría al reto de aprender un nuevo lenguaje de programación, el cual se está convirtiendo en un lenguaje muy popular para el desarrollo en Android.

Tras haber analizado todas estas cuestiones, se debe elegir entre una de las dos opciones, con la que se comenzará a trabajar desde las primeras pruebas que se realicen hasta el final del proyecto. Finalmente, el lenguaje de programación que va a escoger el alumno para este proyecto será Kotlin, debido a los beneficios de este lenguaje y a que, así, el alumno aprenderá un nuevo lenguaje de programación, el cual parece ser una gran apuesta de futuro en este campo.

### Entorno de desarrollo

Para poder desarrollar una app, es necesario elegir también el entorno de desarrollo en el cual se trabajará para poder escribir dicha aplicación. Para ello, al igual que ocurre con los lenguajes de programación, existen muchos entornos de desarrollo distintos, que pueden ser más o menos útiles en función de las características del proyecto que se vaya a realizar.

Lo primero que se va a tener en cuenta en este aspecto es el sistema operativo para el que se va a desarrollar la app. Hay entornos de desarrollo que están destinados al desarrollo de aplicaciones para un sistema operativo concreto, como puede ser el caso de Android Studio, que no permite el desarrollo de aplicaciones híbridas.

El segundo aspecto para tener en cuenta se trata del lenguaje de programación que se va a utilizar. Hay entornos de desarrollo que están destinado a lenguajes concretos [7], por lo que se deben descartar los que estén enfocados para lenguajes que han sido descartados en el apartado anterior.

Así, importantes entornos de desarrollo quedan fuera de la elección del entorno elegido para desarrollar el proyecto, como pueden ser los siguientes: React Native, Apache Cordova, Xamarin y Flutter. Todos estos entornos de desarrollo están enfocados a un lenguaje de programación concreto, por lo que nos limitaría mucho a la hora de trabajar, teniendo en cuenta que ya ha sido elegido Kotlin como el lenguaje de programación definitivo de nuestra app.

Las principales opciones de entornos de desarrollo para un proyecto en Kotlin son IntelliJ IDEA y Android Studio. Este último es, precisamente, el entorno de desarrollo oficial de Google para desarrollar apps para Android, estando basado en IntelliJ IDEA. Se puede entender, por tanto, que Android Studio es un IDE perfecto para el proyecto, al incluir lo mejor de IntelliJ IDEA y ser el IDE oficial de Google.

Considerando que se ha escogido un lenguaje de programación que el alumno no conoce, el reto de escoger un entorno de desarrollo que tampoco domine puede dificultar mucho su trabajo. En este caso, el alumno no ha usado apenas IntelliJ IDEA, mientras que con Android Studio tiene la experiencia de haber cursado una asignatura en la que se trabaja con este IDE.

Por estas razones, el entorno de desarrollo que se va a usar a lo largo del proyecto será Android Studio. Tanto el entorno de desarrollo como el lenguaje de programación son oficiales y soportados por Google, de forma que, en caso de tener problemas con alguno de los dos durante la realización del proyecto, se podrá encontrar documentación y ayuda para su resolución en la red.

### Gestor de base de datos

PROVISIONAL

Uno de los objetivos finales de la app es poder tener un histórico de las mediciones fisiológicas registradas por la red de área corporal a través de su sistema de sensores.

Este requisito no es prioritario dentro del proyecto, ya que hay retos de mayor prioridad e importancia, como poder mostrar en tiempo real los datos recogidos por los sensores; sin embargo, en caso de que el proyecto avance a buen ritmo y se logre resolver esta cuestión con tiempo, podría ser muy interesante tener un registro del histórico del paciente.

Para ello, tras la medición de las señales fisiológicas, los datos se exportarían y almacenarían en una base de datos relacional, en la que existirían, al menos, dos tablas. La primera tabla serviría para almacenar la información de los usuarios que se tomen las mediciones. La segunda tabla, por su parte, podría servir para almacenar las distintas mediciones de los usuarios.

Este formato de base de datos es provisional, ya que se debe definir todavía cómo va a funcionar la app. La decisión final se tomará en base a los requisitos funcionales que se definan en lo que queda de proyecto.

En caso de que se almacene el histórico en una base de datos, se debe tener en cuenta que estos datos no podrán ser guardados en el almacenamiento interno del smartphone, debido al gran tamaño que tendrían, consumiendo mucha memoria. Por tanto, los datos serían almacenados en una base de datos localizada en un servidor en la nube, para lo que habría que buscar la forma óptima de poder guardarlos sin que consuma demasiados recursos.

Esta decisión quedará en el aire hasta que no se conozcan las posibles limitaciones que nos podamos encontrar de recursos, lo cual se probará durante el desarrollo de la app.

PROVISIONAL

### Metodología de trabajo

Además de las tecnologías elegidas para trabajar a lo largo del trabajo, para un mejor control y gestión del trabajo se han elegido una serie de herramientas que no solo faciliten la tarea al alumno, sino que también ayudan a los tutores para realizar un seguimiento más completo quitándoles una menor cantidad de tiempo.

#### Scrum

En primer lugar, aunque no es una tecnología, se puede hablar del marco de trabajo escogido para la realización del trabajo a lo largo del tiempo. Para ello, acordado entre los tutores y el alumno, se ha seguido el marco de trabajo Scrum, dividiendo el proyecto en diferentes sprints en los que el alumno debe ir realizando una serie de tareas para ir completando el proyecto poco a poco.

El comienzo y el final de cada sprint lo definen las reuniones acordadas, en las que se revisa lo realizado durante los días o semanas previos y se ponen nuevas metas a corto plazo. Así, el alumno siempre tiene definido en qué tareas debe trabajar hasta la siguiente reunión, en la que se evaluará el trabajo realizado desde la última reunión.

#### GitHub

Con el objetivo de tener todo el trabajo en un mismo lugar, se ha escogido GitHub para servir de apoyo al alumno durante el trabajo. De esta forma, con GitHub el alumno puede tener un control de versiones muy completo, a través de un repositorio privado que ha sido compartido con sus tutores para poder acceder a todo el trabajo realizado.

Dentro de este repositorio, se pueden encontrar todo el contenido del proyecto (documentos, código, etc.), las distintas tareas acordadas en las reuniones para cada sprint o un gestor de tareas. Todo está organizado de forma que los tutores puedan comprobar en qué situación se encuentra el proyecto del alumno con unos simples clics.

El repositorio tiene asignado un proyecto privado de GitHub, con tres columnas en las que se colocan las distintas tarjetas: To Do (por hacer), In Progress (en progreso) y Done (finalizada). Cada sprint está definido como una Milestone, y sus correspondientes tareas se definen como Issues, cuyo etiquetado ayuda a saber qué tipo de tarea es. De esta forma, se pueden consultar en las Issues los diferentes commits que han tenido, pudiendo acceder a información como los distintos cambios que se han hecho en cada uno.

A la vez que una Issue se crea, en el proyecto interno aparece una tarjeta en la columna To Do, correspondiente a dicha Issue, conteniendo toda su información. Cuando el alumno comience a trabajar en una Issue que se encuentre en la columna To Do, él mismo podrá moverla hacia la columna In Progress, donde se encuentran las tareas que tenga abiertas por estar trabajando en ellas. Finalmente, cuando una Issue es cerrada (con o sin commits), ésta se moverá de forma automática a la columna Done. Así, los tutores podrán observar de una manera sencilla el avance de las tareas en las que el alumno esté trabajando.

El funcionamiento del proyecto interno es muy parecido al de herramientas conocidas como Trello o Planner, pero se ha elegido usar esta opción de GitHub para sacarle un mayor aprovechamiento a todas las ventajas y opciones que permite esta herramienta destinada a desarrolladores.

#### Toggl

En un proyecto de larga duración como es un TFG, al que le corresponden unas 300 horas de trabajo según los créditos asignados, es importante llevar un control de las horas de trabajo y las tareas a las que se han dedicado este tiempo.

Para ello, se ha elegido usar la herramienta Toggl, una herramienta destinada a contabilizar el tiempo que se dedica a cada tarea, para evaluar cómo se ha trabajado en un proyecto. Esta herramienta tiene la posibilidad de crear un proyecto y crear tareas de forma similar a como se identifican en GitHub las Issues, mediante un identificador numérico (por ejemplo, #4 Estado del arte científico).

Una de las ventajas que tiene Toggl es que genera reportes en los que se puede observar el tiempo monitoreado dentro de un marco temporal, desglosándolo en las distintas tareas a las que se ha dedicado el tiempo, además de ver quién ha sido la persona que ha trabajado esas horas. Estos reportes pueden, además, ser descargados tanto en formato PDF como en formato CSV, siendo de gran utilidad para compartirlos con otras personas.

Aprovechando esto último, en el repositorio de GitHub se encuentra una carpeta llamada efforts, en la que el alumno va actualizando los reportes generados por la herramienta Toggl, siendo dos los reportes publicados en el repositorio: por una parte, está el reporte summary.pdf, en el que se resume a cuándo ha correspondido las horas, generando un gráfico para ver cuánto tiempo se ha destinado a cada tarea; por otra parte, está el reporte detailed.pdf, que indica el autor de cada tiempo de trabajo, cuándo ha tenido lugar, a qué tarea corresponde y a qué sprint se corresponde este tiempo.

El alumno ha escogido esta herramienta para el seguimiento de las horas debido a su experiencia previa con ella, conociendo todas las ventajas que ofrece gracias a que tuvo que usarla en una asignatura del Grado. Por ello, sabiendo lo útil que sería para llevar un control de las horas y lo que le facilitaría la tarea a los tutores, entendió que era una herramienta muy interesante para usarla durante la realización del TFG.

#### Overleaf

Un aspecto fundamental en el que se debe trabajar durante todo el proyecto correspondiente al TFG es en la redacción de la memoria. Para ello, se va a usar una herramienta que permita componer textos con una alta calidad tipográfica, usando la tecnología de LaTeX.

De acuerdo entre el alumno y los tutores, la herramienta que se va a usar para este apartado tan importante durante el desarrollo del TFG es Overleaf, debido a sus múltiples ventajas. Entre ellas, se permite trabajar en documentos escritos en LaTeX de manera online, siendo útil para compartir estos documentos y poder trabajar varias personas en ellos en tiempo real, teniendo un control de versiones.

De esta forma, el alumno y los tutores tendrán la memoria en esta plataforma, siendo el alumno el encargado de trabajar en la memoria mediante la plataforma, haciendo un seguimiento los tutores del trabajo que se vaya realizando, ayudando en los posibles problemas que puedan ocurrir en caso de que sea necesario.

La decisión de utilizar la plataforma Overleaf fue tomada tras ser descartados otros editores de texto, buscando obtener un resultado lo más profesional posible a la finalización del proyecto. Así, se pretende conseguir una documentación de mucha calidad, aprovechando una plantilla cedida por el tutor, basada en la plantilla del departamento, cuya estructuración va a respetar el alumno. Esta plantilla destaca por su gran organización y por el resultado tan profesional que ofrece tras la compilación del proyecto, logrando un documento de gran calidad.

#### Herramienta REM

Para la realización de un buen producto, algo indispensable es una buena toma de requisitos. Los requisitos son un conjunto de objetivos desde el punto de vista del usuario que se desean alcanzar con el desarrollo de un software, debiendo ser medibles y teniendo una prioridad a la hora de trabajar en ellos. Así, se podrá comprobar si se han alcanzado o no.

En un marco de trabajo como el que se va a usar en este TFG, en el que vamos a utilizar Scrum, la captura de requisitos no debe ser completa desde el principio. Al estar dividido el proyecto en distintos sprints en los que se evaluará el trabajo realizado, los requisitos pueden ir variando, siendo eliminados unos y añadiendo otros nuevos.

Para un control completo del conjunto de requisitos de este proyecto se va a utilizar una herramienta desarrollada por uno de los departamentos de la Universidad de Sevilla, concretamente el de Lenguajes y Sistemas Informáticos. Esta herramienta se llama REM (de REqueriments Management), siendo una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la Tesis Doctoral “Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información”, presentada por Amador Durán Toro (autor también del software) en septiembre del año 2000. REM se encuentra disponible para descargar en la web del departamento LSI, lanzándose la versión 1.2.2, la más reciente, el 25 de noviembre de 2004, estando destinada para sistemas operativos de Windows.

Este software para toma de requisitos es muy completo. Permite crear documentos de requisitos que pueden ser divididos en distintas secciones, permitiendo la subida también de elementos gráficos. Para una perfecta definición del problema, esta herramienta permite definir organizaciones, reuniones, participantes, objetivos, actores, objetos, valores, etc. Respecto a los propios requisitos, se pueden definir de distintos tipos: de información, de restricción, funcionales y no funcionales. Además, también se pueden crear casos de uso, y analizar que los requisitos se cumplen en los casos de uso con matrices de rastreabilidad.

Se observa, por tanto, que la herramienta REM es un software que puede ser de una gran utilidad en la definición de requisitos del proyecto. Al ser una herramienta propia de uno de los departamentos que imparten clases en el Grado y estando disponible para su descarga gratuita en la propia web del departamento para el uso de desarrolladores, docentes, investigadores, estudiantes y otros, entendemos que puede ser la mejor opción para realizar una completa toma de requisitos para el TFG.

## ***REFERENCIAS***

[1] – **IDC**: Smartphone Market Share.   
<https://www.idc.com/promo/smartphone-market-share>

[2] – **Android Developers Blog**: Android 12 is live in AOSP!   
<https://android-developers.googleblog.com/2021/10/android-12-is-live-in-aosp.html>

[3] – **Android Developers Blog**: The frist developer preview of Android 13.  
<https://android-developers.googleblog.com/2022/02/first-preview-android-13.html>

[4] – **MovilZona.es**: Este sondeo desvela las claves a la hora de cambiar de móvil.  
<https://www.movilzona.es/2020/10/28/sondeo-cuando-como-cambiar-movil/>

\* BUSCAR EL ESTUDIO DE IOINVESTIGACIÓN PARA LA REFERENCIA 4

[5] – **Statcounter Global Stats**: Mobile Android Version Market Share Worldwide  
<https://gs.statcounter.com/android-version-market-share/mobile/worldwide/>

\* BUSCAR LOS DATOS EXACTOS DE GOOGLE PARA LA REFERENCIA 5

[6] – **Statcounter Global Stats**: Mobile & Tablet Android Version Market Share Worldwide <https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide>

\* BUSCAR LOS DATOS EXACTOS DE GOOGLE PARA LA REFERENCIA 6

[7] **El Androide Libre – El Español**: Hacer aplicaciones para Android: ¿Qué es un entorno de desarrollo y cuál es el mejor?  
<https://www.elespanol.com/elandroidelibre/20200518/hacer-aplicaciones-android-entorno-desarrollo-mejor/490952339_0.html>