**COLEGIO SALESIANO SAN JOSÉ**

**CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

****

**PROYECTO FIN DE CICLO**

iParty

Helena Juliá Sánchez

**Febrero, 2020**

**COLEGIO SALESIANO SAN JOSÉ**

**CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**PROYECTO FIN DE CICLO**

iParty

**Autor**: Helena Juliá Sánchez

**Tutor**: María Isabel Martínez García

**Mes, Año**

# Índice

[Índice 1](#_Toc34844224)

[Lista de símbolos 2](#_Toc34844225)

[1. Introducción 4](#_Toc34844226)

[1.1. Objetivos del proyecto 5](#_Toc34844227)

[1.2. Enfoque 5](#_Toc34844228)

[1.2.1. Límites 5](#_Toc34844229)

[1.2.2. Detalles de los objetivos 5](#_Toc34844230)

[2. Antecedentes 6](#_Toc34844231)

[2.1. Aplicaciones para conocer a gente 6](#_Toc34844232)

[2.1.1. Mobifriends 6](#_Toc34844233)

[2.1.2. FriendZone 7](#_Toc34844234)

[2.1.3. MeetMe 8](#_Toc34844235)

[2.2. Buscar mesa 8](#_Toc34844236)

[2.3. Lenguaje de programación: Dart 10](#_Toc34844237)

[2.4. Framework: Flutter 11](#_Toc34844238)

[3. Objetivos del proyecto 15](#_Toc34844239)

[*3.1.* Objetivos generales 15](#_Toc34844240)

[*3.2.* Objetivos específicos 15](#_Toc34844241)

[4. Hipótesis del proyecto 16](#_Toc34844242)

[5. Desarrollo real del proyecto 17](#_Toc34844243)

[5.1. Prototipado de la aplicación 17](#_Toc34844244)

[6. Conclusiones y propuestas 18](#_Toc34844245)

[7. Bibliografía 19](#_Toc34844246)

[8. Planos 21](#_Toc34844247)

[9. Anexos 21](#_Toc34844248)

# Lista de símbolos

# Introducción

El disfrute del tiempo libre se ha convertido en un lujo en el siglo XXI. Y no solo a la hora de disfrutarlo por lo que es, entre responsabilidades y quehaceres. Esto es especialmente cierto cuando esos planes requieren de otras personas. Y más aún cuando el ocio del que pretendes disfrutar es minoritario. Incluso en ciudades grandes, en las que la lógica dice que hay cientos de personas que comparten tus gustos, el anonimato y las distancias resultan de son un impedimento para realizar estos planes.

Es el caso específico de los juegos de rol y de mesa: las editoriales venden, pero muchas personas se encuentran con dificultades para encontrar nuevos grupos con los que realizar su afición. Las opciones más populares para solventarlo pasan por los foros de páginas web anticuadas, como puede ser Comunidad Umbría, o redes sociales enfocadas a otros usos como Facebook o Twitter. Pero no existe ninguna herramienta específica, ágil y enfocada a los usos tecnológicos de esta época.

Precisamente con la idea en mente de crear una aplicación capaz de estar a la altura de las necesidades tecnológicas de los usuarios, se elige el lenguaje de Dart y su framework de Flutter para el desarrollo del proyecto. Flutter permite desarrollar de forma nativa aplicaciones de iOS y Android al mismo tiempo. Se pueden resumir algunas de sus características principales en:

* Compilación rápida y muy personalizable
* Compilación en caliente, que permite visualizar los cambios a medida que se va desarrollando el código
* Desarrollo de la interfaz programática
* Desarrollo de aplicaciones ágiles y eficientes

Para optimizar el rendimiento de la aplicación se emplearán bases de datos no relacionales, utilizando los sistemas proporcionados por Firebase.

## Objetivos del proyecto

El objetivo del proyecto es la creación de una aplicación móvil en la cuál los usuarios se podrán registrar, crear partidas y apuntarse a aquellas que estén abiertas y, en definitiva, conocer a gente con la que compartan gustos. Para ello se utilizará tecnología puntera en el desarrollo de estas: el framework de Google Flutter basado en el lenguaje Dart.

A pesar de que Flutter permite el desarrollo de aplicaciones nativas para iOS y Android y su objetivo pasa porque estas aplicaciones se vean naturales dentro del diseño único y característico de ambos dispositivos, la imposibilidad material de compilar para iOS implica que en este proyecto no se va a lograr este fin. La aplicación funcionará tanto para Android como para iOS, pero su aspecto será más parecido a los estándares de Android que los de iOS.

## Enfoque

El enfoque del proyecto es el siguiente:

### Límites

A pesar de que tanto el lenguaje de Dart como el framework de Flutter pertenecen a Google y Google ha puesto un gran empeño en documentar y popularizarlos, son dos tecnologías relativamente jóvenes y con pocos adeptos. Existen pocos recursos en la red, exceptuando los oficiales (que en ocasiones resultan parcos a la hora de dar ejemplos de uso), que podrían dificultar el desarrollo del proyecto.

### Detalles de los objetivos

Los objetivos del proyecto son los siguientes:

* Aprender y familiarizarme con el lenguaje Dart y el framework de Flutter.
* El desarrollo de una app móvil que permita a los distintos usuarios comunicarse y conocerse en base a unos gustos comunes.

# Antecedentes

## Aplicaciones para conocer a gente

Las aplicaciones para conectar a gente no son muy populares y existen pocos ejemplos funcionales cuyo único objetivo no sea otro que el de ligar. Aun así, existen algunos ejemplos de desarrollo que pueden servir de ejemplo y de qué se quiere conseguir y qué no se quiere conseguir al realizar la aplicación.

Estas aplicaciones tienen algunos elementos comunes, que se verán a continuación y que pueden resumirse en:

* Listado de usuarios
* Chat
* Perfil de usuario

### Mobifriends

Se trata de una aplicación que busca conectar a gente, ya sea con intereses románticos o para buscar amistad. Según inicias sesión, muestra un listado de usuarios que se encuentran en tu zona.

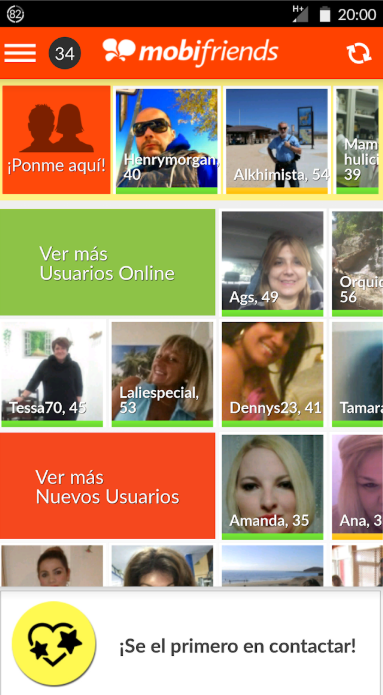
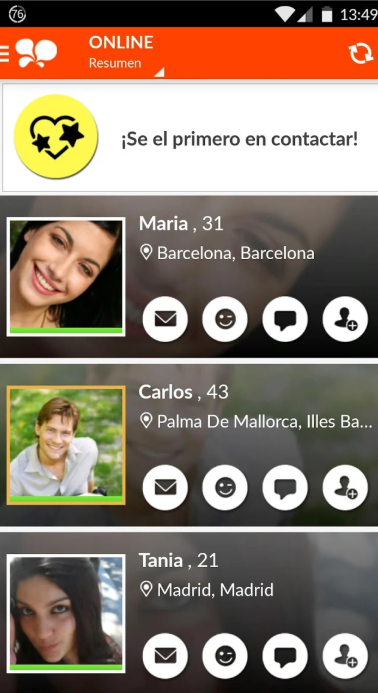
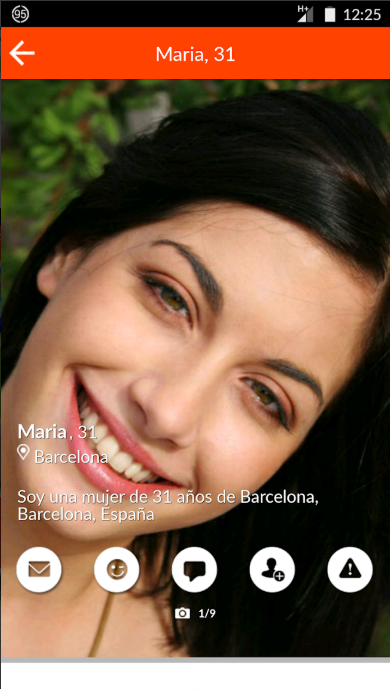
   

Ilustración 1: Imágenes promocionales de la app Mobifriends (Mobifriends, 2019)

La aplicación cuenta con varias páginas que permiten buscar entre usuarios aplicando diversos filtros (como el de localización), ver su perfil y contactar con ellos. También dispone de un apartado de personas favoritas que permite acceder fácilmente a aquellos usuarios que hayamos marcado como tal.

Se aprecian las siguientes características, tal y como se resume en la Tabla 1.

Tabla 1: Pros y contras de Mobifriends

|  |  |
| --- | --- |
| PROS | CONTRAS |
| * Filtros entre usuarios * Contacto de usuarios * Usuarios favoritos | * Interfaz anticuada y fea * Orientada especialmente a las interacciones románticas * Es una página de contactos, no un portal para buscar grupos de amigos |

### FriendZone

FriendZone es una app cuyo enfoque principal es unir gente a partir de sus intereses, por lo que se parece mucho más al objetivo de desarrollo de iParty.

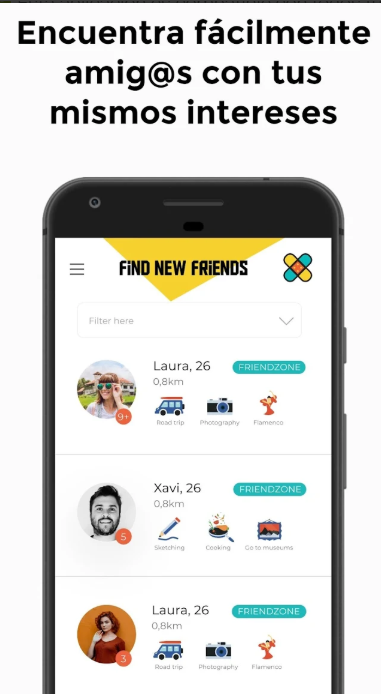
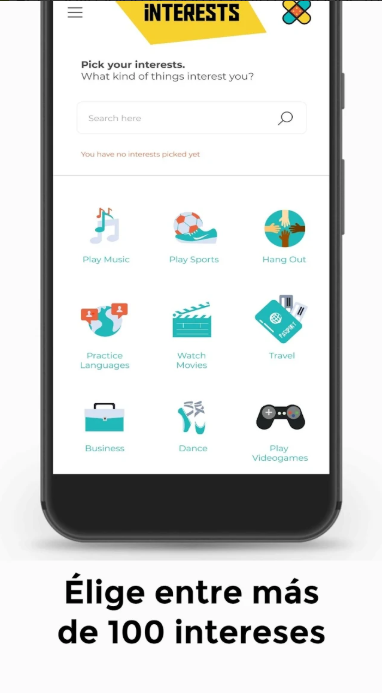
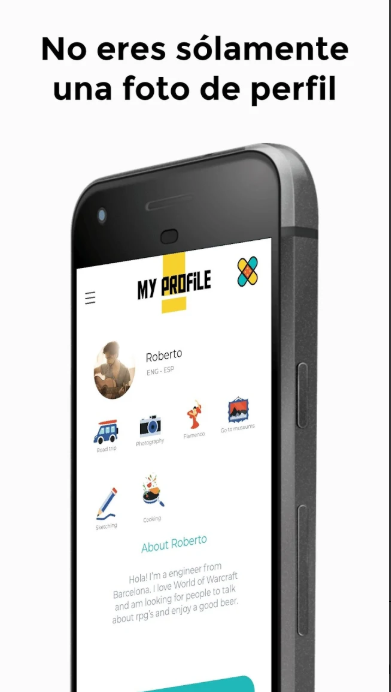
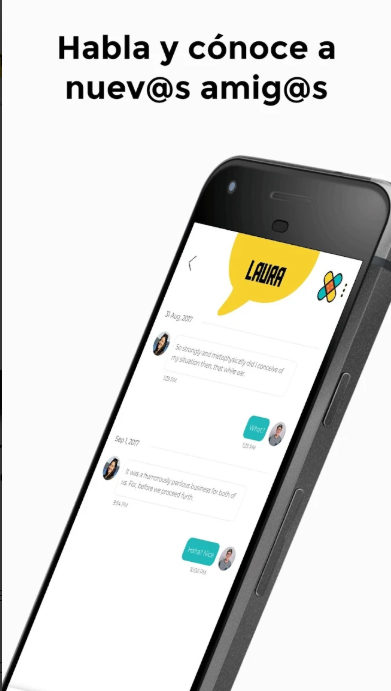
   

Ilustración 2: Imágenes promocionales de FriendZone (FriendZone, 2020)

El punto de interés en el desarrollo de la aplicación radica en los gustos de cada uno, poniendo especial interés en las aficiones de cada uno. La aplicación permite filtrar a la gente en grupos y subgrupos (por ejemplo, películas > terror) y distancia y dispone de un algoritmo para encontrar a posibles amigos.

Se aprecian las siguientes características, resumidas en la Tabla 2.

Tabla 2: Pros y contras de FriendZone

|  |  |
| --- | --- |
| PROS | CONTRAS |
| * Interfaz bonita y moderna | * Demasiado genérica |

### MeetMe

MeetMe se vende como una aplicación para buscar amigos con los que chatear.

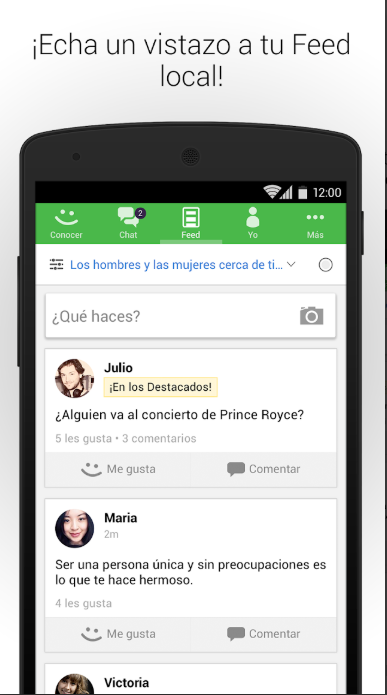
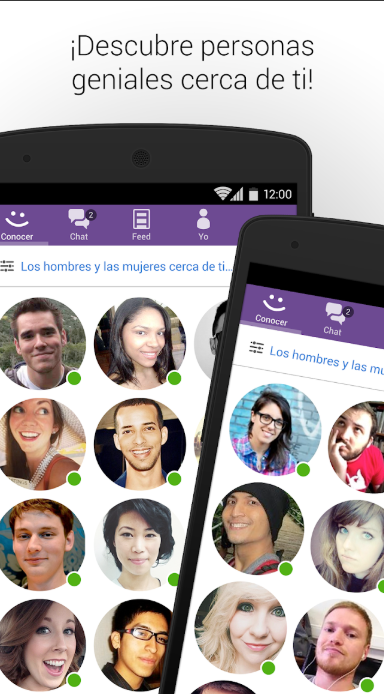
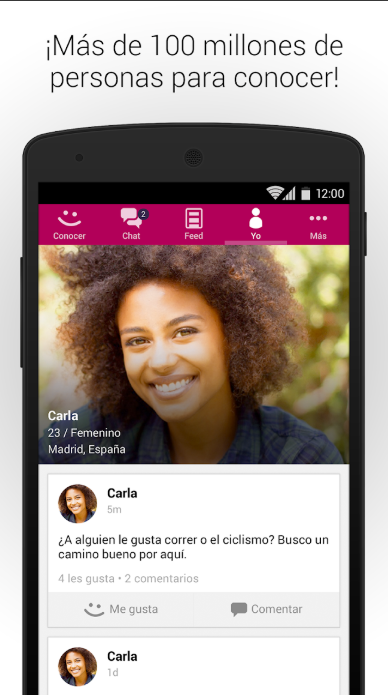
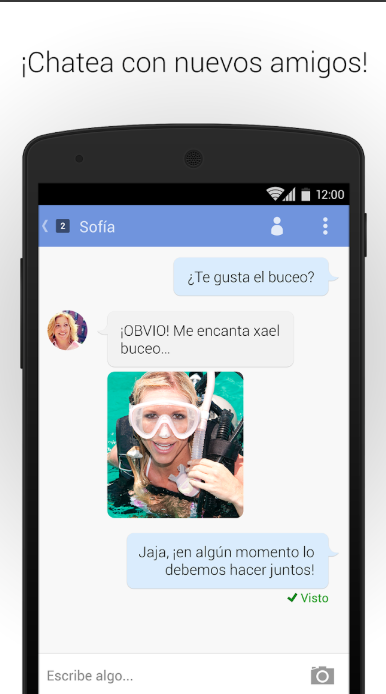
  

Ilustración 3: Imágenes promocionales de MeetMe (MeetMe.com, 2020)

Además del chat, dispone de un *feed* que recuerda al muro de Facebook. Se pueden ver sus características principales en la Tabla 3.

Tabla 3: Pros y contras de MeetMe

|  |  |
| --- | --- |
| PROS | CONTRAS |
| * Navegación intuitiva | * Sin objetivo claro * Más red social que punto de encuentro * No da información clara sobre sus usuarios * Sin filtros claros de búsqueda |

## Buscar mesa

Si las opciones a la hora de conocer a gente a través de aplicaciones son limitadas, estas empeoran cuando tu objetivo es conocer a gente para jugar a rol y a juegos de mesa. La única aplicación de las analizadas anteriormente (FriendZone) que dispone de filtros sobre aficiones no tiene los adecuados para nuestro objetivo y se centran más en conocer a gente que en realizar una actividad.

Es por eso por lo que normalmente se usan otros canales que no siempre resultan óptimos para la búsqueda. Un buen ejemplo es la Ilustración 4, en la que podemos ver un llamamiento de una mesa que busca jugadores:



Ilustración 4: Ejemplo de búsqueda de jugadores en Twitter (En5MinutosRol, 2020)

En5MinutosRol hace así un llamamiento en Twitter utilizando como únicos medios de difusión:

* Su cuenta
* La cuenta de una editorial de juegos de rol

Este tipo de comunicación, por llamamiento, puede resultar óptima si consigue popularizarse a través de los retweets. Sin embargo, también puede acabar en el olvido sin conseguir ninguna clase de respuesta. Aunque haya jugadores que quieran jugar y una partida organizada con tal fin.

Otra opción popularizada a la hora de buscar grupos de rol es acudir a foros especializados, en los que los usuarios pueden publicar mensajes como el de la Ilustración 5:

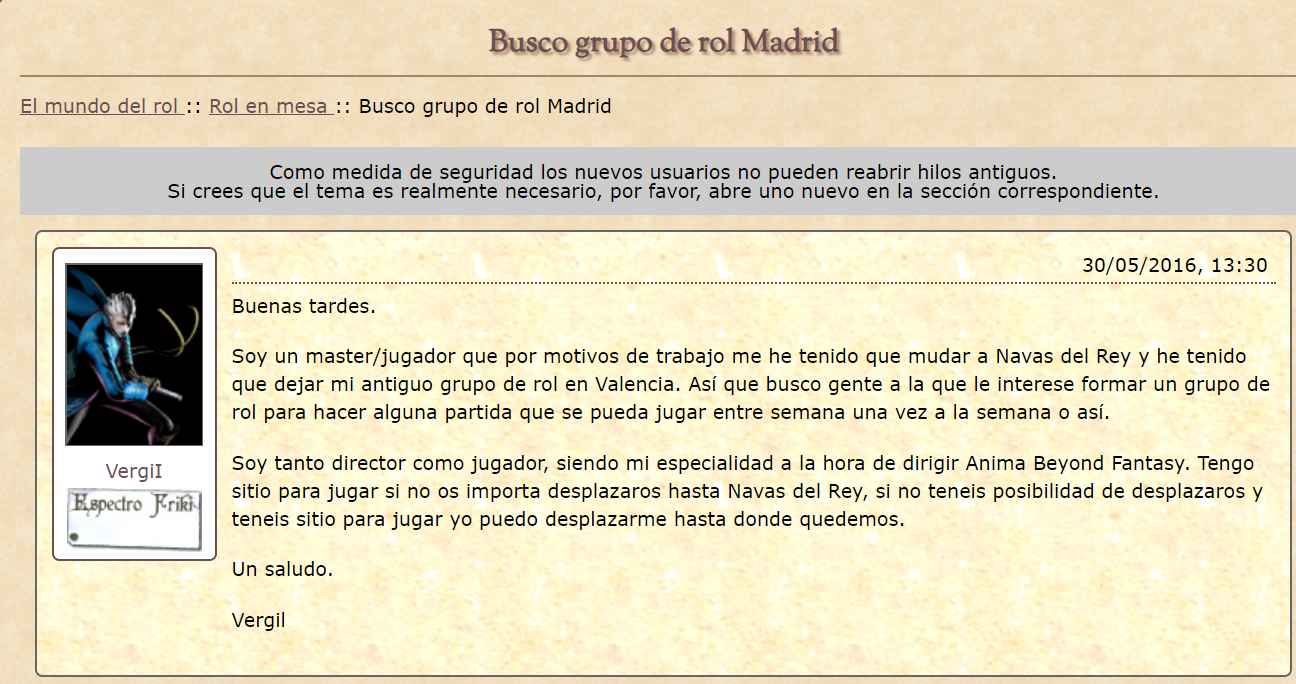


Ilustración 5: Mensaje buscando grupo de rol en Madrid (VergiI, 2016)

De nuevo, este tipo de opciones no resultan óptimas debido a que no es su objetivo fundamental. Es un foro, en el que los mensajes pueden quedar fácilmente tapados por otros mensajes más populares.

## Lenguaje de programación: Dart

Dart es desarrollado por Google. Se trata de un lenguaje dinámico e inspirado en JavaScript. Además, está influido por Java y C# (Oppong, 2019). Se desarrolló en 2011 y, por aquella época, su característica principal se basaba en que estaba orientado a objetos y basado en el uso de clases (algo de lo que JavaScript, su principal competencia, carecía). Sin embargo, y aunque era el objetivo de Google (fue bautizado como el *JavaScript Killer[[1]](#footnote-1)* en la red), no fue capaz de superarlo y en 2013, con la aparición de TypeScript, fue superado.

Aun así, Dart sigue desarrollándose. Y, todo en cuenta, tiene ciertos puntos fuertes que caben destacar (Swehli, 2018):

* Tiene una sintaxis sencilla y basada en C, por lo que resulta muy similar a la que se pueden encontrar en lenguajes como C++, Java y JavaScript (Bolton, 2019)
* Es un lenguaje rápido, eficiente[[2]](#footnote-2) y *libre de sorpresas* (Dart, s.f.)
* Dispone de muchísimas librerías para facilitar su desarrollo
* Es multidisciplinar: está enfocado al desarrollo móvil y web, de aplicaciones de escritorio y de servidores
* Es capaz de compilar de manera nativa en JavaScript
* Está desarrollado por Google

A pesar de su impopularidad, desde Google se sigue desarrollando en Dart. Como ejemplos de desarrollo web están Google AdWords y Google AdSense (AngularDart, s.f.), que el framework de AngularDart. Fuchsia, otro proyecto de Google, también está desarrollado en Dart. Se trata de un sistema operativo *open-source* que ha sido introducido en la gama de productos de Google Home Hub (Bolton, 2019).

## Framework: Flutter

Flutter es un framework de Dart desarrollado por Google en 2017. Con él se pueden desarrollar aplicaciones para cualquier dispositivo móvil, ya sea iOS o Android, con una interfaz unificada y sin errores y manteniendo la velocidad y el rendimiento de las aplicaciones nativas (Bolton, 2019). Y, lo más importante, manteniendo un único código fuente (Thomas, 2019). Además, se pretende que estas aplicaciones no solo funcionen en cualquier entorno, sino que resulte natural en las diferentes plataformas (Flutter, s.f.).

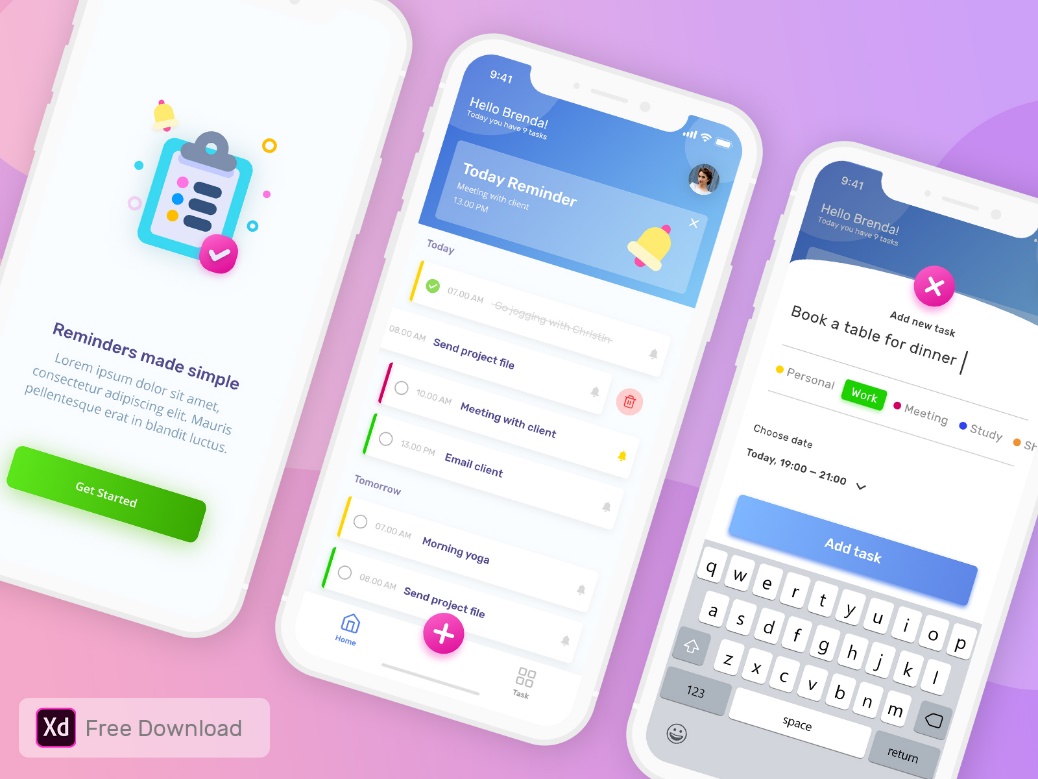


Ilustración 6: To do list es un ejemplo de aplicación que se puede desarrollar en Flutter (LiveLikeCounter, 2019)

A pesar de que no se trata de una tecnología popularizada en el desarrollo de aplicaciones, Google sí que la utiliza para desarrollar. Gracias a esto, el equipo de Flutter no para de actualizar los componentes para asegurarse de que estos están al día y mantiene una documentación extensiva y que recoge cada uno de sus elementos (Concise Software, 2019).

El núcleo de Flutter se trata de los widgets: prácticamente todos los elementos que nos encontramos en el desarrollo de aplicaciones lo van a ser. Estemos hablando de columnas y filas o imágenes y texto, se tratan del mismo elemento adaptado a las necesidades específicas de cada momento. Así se crean las pantallas y componentes en Flutter: anidando widgets en lo que se llama *widget tree*. Un ejemplo del diseño de una visualización se puede ver en la Ilustración 7:

Ilustración 7: Ejemplo de widget tree en Flutter

El esquema anterior es una posible interpretación de la siguiente pantalla, pudiendo conseguirse el mismo resultado utilizando otra combinación de widgets:

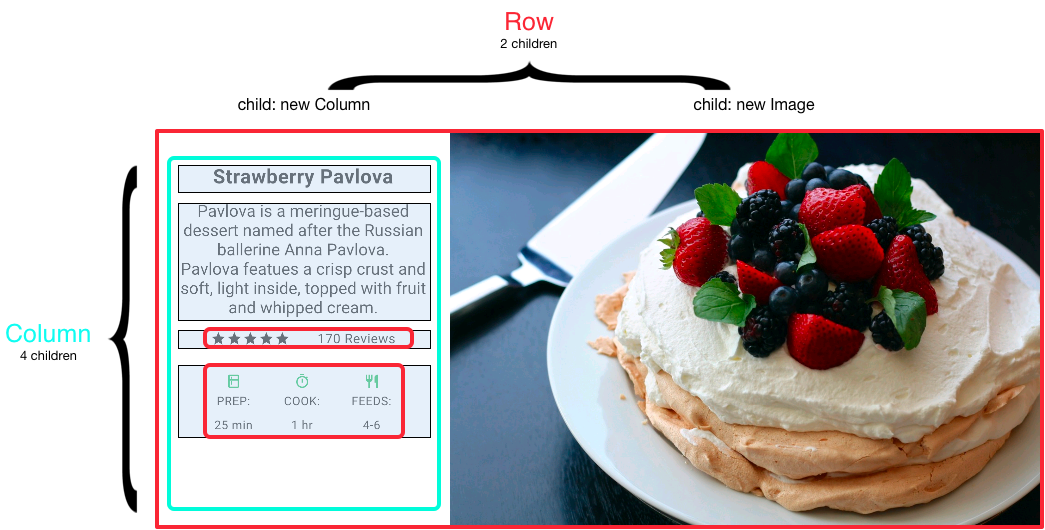


Ilustración 8: Diseño de una pantalla en Flutter (Layouts in Flutter, s.f.)

Algunas de las características de Flutter son (Bolton, 2019):

* Compilación Ahead-of-Time (AOT), lo que permite que la aplicación sea capaz de lanzarse de manera rápida
* Compilación Just-in-Time (JIT), que es la que permite la *hot reload*, es decir, la visualización de cambios de la aplicación en tiempo real
* Muestra entre 60 y 120 imágenes por segundo, si el dispositivo lo permite

Para conseguir un aspecto natural para iOS y para Android, Flutter utiliza dos librerías: cupertino y material. A continuación, se muestra esta dualidad adaptativa a partir de la misma pantalla de una aplicación de ejemplo orientadas a cada una de estas plataformas:

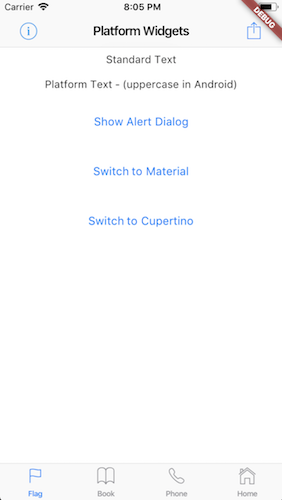
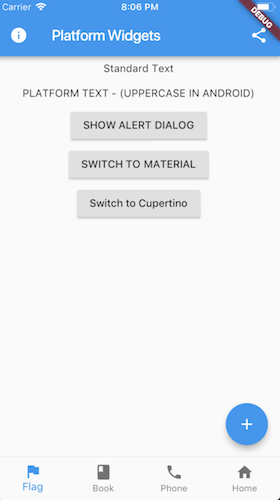
 

Ilustración 9: Comparativa entre la librería de cupertino (para iOS) y material (para Android)

# Objetivos del proyecto

Los objetivos que se plantean en el desarrollo del siguiente TFC son:

## Objetivos generales

A nivel general, se plantea como objetivo la realización de una aplicación móvil para conectar a gente que comparta aficiones de juegos de mesa y rol.

## Objetivos específicos

Los objetivos específicos del TFC son los siguientes:

1. Permitir el registro y acceso de usuarios.
2. Crear mesas de juego que busquen jugadores, pudiendo agregar comentarios y formas de contacto.
3. Realizar búsquedas de mesas de juego o contactos empleando de filtros.
4. Aprender las tecnologías de Flutter y Dart, así como familiarizarme con el proveedor de servicios de Firebase y su forma de interactuar con aplicaciones móviles.
5. Aunar los conocimientos adquirido a lo largo de toda la formación de DAM, empleando para ello las siguientes competencias: planificación de proyectos, desarrollo de bases de datos no relacionales, procesos, programación orientada a objetos y arquitectura y análisis del software.

# Hipótesis del proyecto

# Desarrollo real del proyecto

## Prototipado

Se desarrolla un prototipado de iParty a partir de la herramienta Quick Proto, que permite diseñar y enlazar de manera intuitiva diferente pantallas (O'Brien, 2019). Se pretende tener una sensación general de las diferentes secciones que va a contener iParty, así como la interacción que tendrá con sus diferentes componentes.

Imagen que contiene texto

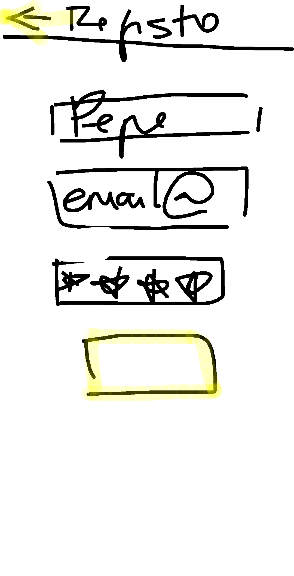
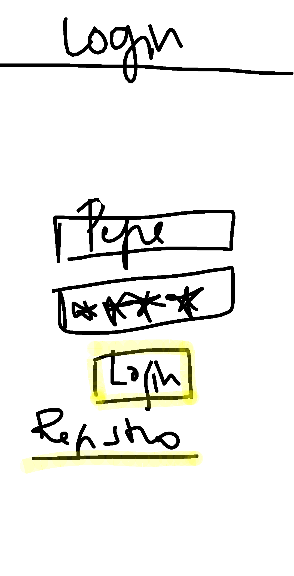
Descripción generada automáticamente

Ilustración 10: iParty, diseño de inicio

## Elección de la paleta de colores

# Conclusiones y propuestas

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Bibliografía

*AngularDart*. (s.f.). Obtenido de Google Open Source: https://opensource.google/projects/angulardart

Bolton, D. (27 de marzo de 2019). *The Fall and Rise of Dart, Google’s ‘JavaScript Killer’*. Obtenido de Dice: https://insights.dice.com/2019/03/27/fall-rise-dart-google-javascript-killer/

Concise Software. (26 de agosto de 2019). *What is Flutter? Here is everything you should know*. Obtenido de Medium: https://medium.com/@concisesoftware/what-is-flutter-here-is-everything-you-should-know-faed3836253f

*Dart*. (s.f.). Obtenido de Google Open Source: https://opensource.google/projects/dart

*Dart vs JavaScript*. (5 de marzo de 2019). Obtenido de codemagic: https://blog.codemagic.io/dart-vs-javascript/

En5MinutosRol. (10 de marzo de 2020). *En 5 Minutos Rol*. Obtenido de Twitter: https://twitter.com/En5MinutosRol/status/1237373767661338625?s=09

Firebase. (s.f.). *Firebase*. Obtenido de Firebase: https://firebase.google.com/?hl=es

*Flutter*. (s.f.). Obtenido de Google Open Source: https://opensource.google/projects/flutter

Flutter. (s.f.). *Flutter*. Obtenido de Flutter: https://flutter.dev/

FriendZone, A. T. (4 de marzo de 2020). *FriendZone*. Obtenido de PlayStore: https://play.google.com/store/apps/details?id=com.app.friendzone

inVerita. (10 de marzo de 2020). *Flutter vs Native vs React-Native: Examining performance*. Obtenido de Medium: https://medium.com/swlh/flutter-vs-native-vs-react-native-examining-performance-31338f081980

*Layouts in Flutter*. (s.f.). Obtenido de Flutter: https://flutter.dev/docs/development/ui/layout

LiveLikeCounter. (10 de diciembre de 2019). *Flutter-Todolist*. Obtenido de GitHub: https://github.com/LiveLikeCounter/Flutter-Todolist

MeetMe.com. (24 de febrero de 2020). *MeetMe*. Obtenido de PlayStore: https://play.google.com/store/apps/details?id=com.myyearbook.m

Mobifriends. (16 de diciembre de 2019). *Mobifriends* . Obtenido de PlayStore: https://play.google.com/store/apps/details?id=com.mobifriends.app

O'Brien, D. (3 de enero de 2019). *Quick Proto*. Obtenido de Google Play: https://play.google.com/store/apps/details?id=net.dobrien.quickproto

Oppong, J. (11 de marzo de 2019). *Why Dart?* Obtenido de Creative Bracket: https://creativebracket.com/why-dart/

Ramirez, V. (13 de febrero de 2017). *Qué es Firebase de Google: 7 motivos para usar esta plataforma en tu app*. Obtenido de Marketing 4 ecommerce: https://marketing4ecommerce.net/

Swehli, M. (12 de marzo de 2018). *Why Dart is the Language to Learn of 2018*. Obtenido de Medium: https://medium.com/@mswehli/why-dart-is-the-language-to-learn-of-2018-e5fa12adb6c1

Thomas, G. (12 de diciembre de 2019). *What is Flutter and Why You Should Learn it in 2020*. Obtenido de freeCodeCamp: https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/

Vega, C. (5 de marzo de 2019). *¿Por qué Flutter usa Dart?* Obtenido de Medium: https://medium.com/

VergiI. (30 de marzo de 2016). *Busco grupo de rol Madrid*. Obtenido de Comunidad Umbria: https://www.comunidadumbria.com/comunidad/foros/tema/41863

waldestrand. (5 de julio de 2008). *Fases del Proceso de Desarrollo del Software*. Obtenido de Sistemas VD: https://sistemasvd.wordpress.com/

# Planos

# Anexos

1. Tal y como señala Bolton en su artículo para Dice (2019). [↑](#footnote-ref-1)
2. En pruebas de rendimiento, llega a duplicar la velocidad de procesamiento de JavaScript (Dart vs JavaScript, 2019). Cuando se habla de desarrollo móvil, mantiene un rendimiento muy similar al de Swift/Objetive-C y Java/Kotlin en pruebas de rendimiento Gauss-Legendre (rendimiento de memoria). En pruebas con el algoritmo Borwein (rendimiento de CPU), Flutter se posiciona entre 2 (Android) y 5 (iOS) veces más lento (inVerita, 2020). [↑](#footnote-ref-2)