

# DicereGram



Grado en Ingeniería Informática

## Trabajo Fin de Grado

Autor:

Paula Frías Arroyo

Tutor/es:

Domingo Gallardo López

Junio 2018



Universitat d'Alacant  
Universidad de Alicante

# Contexto

---

La iniciativa de este proyecto surge debido a la inexistencia de aplicaciones móviles de mensajería diseñadas para personas invidentes o con discapacidad visual.

Actualmente, la cifra estimada de personas con discapacidad visual es de 285 millones, de las cuales 39 millones son personas con ceguera absoluta. Un grupo de expertos<sup>1</sup> prevé que esta cifra se triplique en 2050. Esto junto con la escasa accesibilidad actual de las grandes aplicaciones móviles plantea un gran problema tecnológico a solventar.

Atendiendo a lo expuesto en los dos párrafos anteriores, y puesto que realizar un proyecto de ayuda a un colectivo con discapacidad era uno de mis mayores propósitos, emerge la idea de aunar el uso de la API de una de las grandes aplicaciones de mensajería móvil junto con la implementación de una aplicación móvil basada en el reconocimiento del habla.

Con el objetivo de acercar tecnología y comunicación a personas invidentes o con discapacidad visual, nace DicereGram.

---

<sup>1</sup> The Lancet Global Health [Volume 5 No. 9](#)

# Agradecimientos

---

Ante todo, me gustaría expresar mi agradecimiento a todas aquellas personas que, directa o indirectamente, me han apoyado y alentado durante todo el proceso de desarrollo del Trabajo Final de Carrera. Esto no habría podido ser posible sin ninguno de ellos.

En primer lugar, agradecer a mis padres, mis hermanos, mi pareja y mis amigos por todo el apoyo durante estos largos años, por aguantar mis vaivenes emocionales debido a los numerosos problemas encontrados durante el desarrollo del proyecto y por ser, sin lugar a dudas, mi mayor pilar durante todo el camino universitario.

Agradecer también a mi tutor, Domingo Gallardo López, por aceptar mi proyecto y por ser el mejor guía que habría podido tener. Por animarme e instigarme a hacerlo mejor en cada una de las reuniones. Por los días en los que llegaba muy desanimada a la reunión y terminaba llorando de la risa. Por todo, desde LPP, pasando por MADS y terminando por el TFG, gracias.

Por último, dar las gracias a la Universidad de Alicante, a la Escuela Politécnica Superior y a todos los docentes que me han impartido clase durante estos años, con los que, además de mejorar mis conocimientos, he reafirmado cuál es mi vocación, todo lo que conlleve la informática.

“Creativity is contagious, pass it on” – Albert Einstein

# Índice

---

CONTEXTO .....	1
AGRADECIMIENTOS .....	2
ÍNDICE.....	3
ÍNDICE DE ILUSTRACIONES.....	7
1. INTRODUCCIÓN.....	10
2. MARCO TEÓRICO O ESTADO DEL ARTE.....	11
2.1. APLICACIONES MÓVILES .....	11
2.1.1. BLIND COMMUNICATOR.....	11
2.1.2. DEAF-BLIND COMMUNICATOR.....	11
2.1.3. MOBILITY LAUNCHER.....	12
2.1.4. BLIND SMS READER 3.2.....	13
2.1.5. BLIND NAVIGATOR .....	13
2.1.6. COMAID .....	14
2.2. ESTUDIOS PREVIOS RELACIONADOS .....	15
2.2.1. DEVELOPING APPS FOR VISUALLY IMPAIRED PEOPLE: LESSONS LEARNED FROM PRACTICE [49TH HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES 2016]... 15	
2.2.1.1. Introducción.....	15
2.2.1.2. Entrevistas formativas .....	15
2.2.1.3. Desarrollo y diseño de prototipos.....	15
2.2.1.4. Estudio de usuario .....	16
a) Procedimiento .....	16
b) Resultados .....	16
c) Conclusiones .....	17
2.3. REFLEXIONES.....	17
3. OBJETIVOS.....	18
4. METODOLOGÍA .....	19
4.1. SCRUM Y KANBAN .....	19
4.2. GITFLOW .....	19
4.3. TRELLO: TABLERO .....	21

4.3.1.	LISTAS .....	21
4.3.2.	TARJETAS .....	23
4.4.	CONTROL DE VERSIONES .....	24
4.5.	RELEASES .....	25
5.	TECNOLOGÍAS .....	27
5.1.	XAMARIN - VISUAL STUDIO 2017 .....	27
5.2.	SQLITE .....	27
5.3.	ANDROID TTS .....	28
5.4.	ANDROID SPEECH RECOGNIZER.....	29
5.5.	ANDROID GESTURE DETECTOR .....	32
5.6.	TLSHARP.....	34
5.7.	TELEGRAM.....	34
6.	ARQUITECTURA E IMPLEMENTACIÓN .....	35
6.1.	PRIMERAS APROXIMACIONES .....	35
6.1.1.	KOTLIN .....	35
6.1.1.1.	Introducción .....	35
6.1.1.2.	Implementación .....	35
6.1.1.3.	Conclusión.....	37
6.1.2.	API REST EN C# .....	37
6.1.2.1.	Introducción .....	37
6.1.2.2.	Implementación .....	37
6.1.2.3.	Conclusión.....	39
6.1.3.	APIs DE RECONOCIMIENTO DE VOZ .....	39
6.1.3.1.	DialogFlow .....	39
6.1.3.2.	WIT.AI.....	40
6.1.3.3.	Google Speech.....	41
6.1.3.4.	LUIS (Language Understanding Intelligent Service) .....	41
6.1.3.5.	Conclusiones .....	42
6.2.	DIAGRAMA ENTIDAD-RELACIÓN.....	42
6.3.	IMPLEMENTACIÓN Y CONTROL DE CONVERSACIONES.....	43
6.4.	INTERFACES .....	45

6.5.	SERVICIOS .....	45
6.6.	REPOSITARIOS.....	50
6.7.	SERVICIOS EN SEGUNDO PLANO.....	56
7.	DISEÑO .....	57
7.1.	LOGO Y MARCA.....	57
7.2.	PALETA DE COLORES .....	58
7.3.	MOCKUPS.....	60
7.3.1.	MOCKUP LOGIN Y REGISTRO .....	60
7.3.2.	MOCKUP MENÚ PRINCIPAL .....	61
7.3.3.	MOCKUP ENVIAR MENSAJE .....	62
7.3.4.	MOCKUP LEER MENSAJES VISTA GENERAL.....	63
7.3.5.	MOCKUP LEER MENSAJES VISTA INDIVIDUAL .....	64
7.3.6.	MOCKUP MENÚ CONTACTOS.....	65
7.3.7.	MOCKUP BLOQUEO/DESBLOQUEO CONTACTOS .....	66
7.3.8.	MOCKUP MENÚ CONFIGURACIÓN .....	67
7.3.9.	CONCLUSIÓN.....	68
7.4.	DIAGRAMA DE NAVEGACIÓN .....	69
8.	FUNCIONALIDADES .....	70
8.1.	DIAGRAMA DE CASOS DE USO.....	70
8.2.	FUNCIONALIDADES .....	70
8.3.	INTERACCIÓN CONVERSACIONES.....	73
8.3.1.	REGISTRO Y LOGIN .....	74
8.3.2.	MENÚ PRINCIPAL .....	75
8.3.3.	ENVIAR MENSAJE .....	76
8.3.4.	MENÚ CONTACTOS .....	77
8.3.5.	BUSCAR CONTACTO.....	78
8.3.6.	BLOQUEAR/DESBLOQUEAR UN CONTACTO .....	79
8.3.7.	MENSAJES GENERAL .....	80
8.3.8.	MENSAJES CHAT INDIVIDUAL .....	81
8.3.9.	MENÚ CONFIGURACIÓN.....	82
9.	SEGURIDAD .....	83

9.1.	MTPROTO.....	83
9.2.	GDPR (GENERAL DATA PROTECTION REGULATION).....	84
10.	TESTING .....	86
10.1.	NUNIT.....	86
10.1.1.	TIPO DE PRUEBAS UNITARIAS REALIZADAS .....	86
10.1.2.	VISTA Y EJECUCIÓN EN XAMARIN .....	86
10.1.3.	TEST UNITARIOS.....	88
11.	CÓDIGO E INSTALACIÓN.....	93
12.	CONCLUSIONES Y MEJORAS .....	100
13.	BIBLIOGRAFÍA Y REFERENCIAS .....	101
14.	ANEXOS .....	103
14.1.	INTERACCIÓN CONVERSACIONES.....	103
14.1.1.	MENÚ PRINCIPAL .....	103
14.1.2.	ENVIAR MENSAJE .....	104
14.1.3.	MENÚ CONTACTOS .....	105
14.1.4.	BUSCAR CONTACTO.....	108
14.1.5.	BLOQUEAR/DESBLOQUEAR UN CONTACTO .....	109
14.1.6.	MENSAJES GENERAL .....	111
14.1.7.	MENSAJES CHAT INDIVIDUAL .....	113
14.1.8.	MENÚ CONFIGURACIÓN.....	115

# Índice de ilustraciones

---

<i>Ilustración 1 - Logo e interfaz Blind Communicator</i> .....	11
<i>Ilustración 2 - Logo e interfaz Deaf-Blind Communicator</i> .....	11
<i>Ilustración 3 - Logo Mobility Launcher</i> .....	12
<i>Ilustración 4 - Interfaz Mobility Launcher</i> .....	12
<i>Ilustración 5 - Logo e interfaz Blind SMS Reader</i> .....	13
<i>Ilustración 6 - Logo Blind Navigator</i> .....	13
<i>Ilustración 7 - Interfaz Blind Navigator</i> .....	13
<i>Ilustración 8 - Logo Comaid</i> .....	14
<i>Ilustración 9 - Interfaz Comaid</i> .....	14
<i>Ilustración 10 - Interfaz All Apointments</i> .....	16
<i>Ilustración 11 - Conclusiones All Apointments</i> .....	17
<i>Ilustración 12 - GitFlow</i> .....	20
<i>Ilustración 13 - Listas Trello DicereGram</i> .....	22
<i>Ilustración 14 - Ejemplo tarjeta Trello DicereGram</i> .....	23
<i>Ilustración 15 - Gráfica 1 commits DicereGram</i> .....	24
<i>Ilustración 16 - Gráfica 2 commits DicereGram</i> .....	24
<i>Ilustración 17 - Ramas GitHub</i> .....	24
<i>Ilustración 18 - Logo Xamarin</i> .....	27
<i>Ilustración 19 - Logo Visual Studio 2017</i> .....	27
<i>Ilustración 20 - Logo SQLite</i> .....	27
<i>Ilustración 21 - Lista Terminados Trello proyecto Kotlin</i> .....	36
<i>Ilustración 22 - Gráfica commits proyecto Kotlin</i> .....	36
<i>Ilustración 23 - Lista Terminados Trello proyecto API Rest</i> .....	38
<i>Ilustración 24 - Gráfica commits proyecto API Rest</i> .....	38
<i>Ilustración 25 - Logo Api.ai</i> .....	39
<i>Ilustración 26 - Logo DialogFlow</i> .....	39
<i>Ilustración 27 - Tabla de precios DialogFlow</i> .....	40
<i>Ilustración 28 - Logo Wit.ai</i> .....	40
<i>Ilustración 29 - Tabla de precios API Google Speech</i> .....	41



<i>Ilustración 30 - Tabla de precios LUIS .....</i>	<i>42</i>
<i>Ilustración 31 - Diagrama Entidad Relación DicereGram .....</i>	<i>43</i>
<i>Ilustración 32 - Diagrama arquitectura DicereGram .....</i>	<i>43</i>
<i>Ilustración 33 - Diagrama capa servicios DicereGram .....</i>	<i>45</i>
<i>Ilustración 34 - Guía de color marketing .....</i>	<i>57</i>
<i>Ilustración 35 - Logo DicereGram .....</i>	<i>58</i>
<i>Ilustración 36 - Paleta de colores DicereGram.....</i>	<i>58</i>
<i>Ilustración 37 - Mockups login y registro.....</i>	<i>61</i>
<i>Ilustración 38 - Mockups menú principal.....</i>	<i>62</i>
<i>Ilustración 39 - Mockups enviar mensaje .....</i>	<i>63</i>
<i>Ilustración 40 - Mockups mensajes general .....</i>	<i>64</i>
<i>Ilustración 41 - Mockup mensajes individual.....</i>	<i>65</i>
<i>Ilustración 42 - Mockups menú contactos .....</i>	<i>66</i>
<i>Ilustración 43 - Mockups bloqueo/desbloqueo contactos.....</i>	<i>67</i>
<i>Ilustración 44 - Mockups menú configuración .....</i>	<i>68</i>
<i>Ilustración 45 - Diagrama de navegación DicereGram .....</i>	<i>69</i>
<i>Ilustración 46 - Diagrama de casos de uso DicereGram .....</i>	<i>70</i>
<i>Ilustración 47 - Conversaciones registro y login .....</i>	<i>74</i>
<i>Ilustración 48 - Conversaciones menú principal.....</i>	<i>75</i>
<i>Ilustración 49 - Conversaciones enviar mensaje .....</i>	<i>76</i>
<i>Ilustración 50 - Conversaciones menú contactos .....</i>	<i>77</i>
<i>Ilustración 51 - Conversaciones buscar contacto .....</i>	<i>78</i>
<i>Ilustración 52 - Conversaciones bloquear/desbloquear contacto .....</i>	<i>79</i>
<i>Ilustración 53 - Conversaciones mensajes general.....</i>	<i>80</i>
<i>Ilustración 54 - Conversaciones chat individual.....</i>	<i>81</i>
<i>Ilustración 55 - Conversaciones menú configuración .....</i>	<i>82</i>
<i>Ilustración 56 - Estructura MTPProto .....</i>	<i>83</i>
<i>Ilustración 57 - Logo NUnit .....</i>	<i>86</i>
<i>Ilustración 58 - Visual Studio 2017 .....</i>	<i>87</i>
<i>Ilustración 59 - Vistas ejecución test unitarios NUnit .....</i>	<i>88</i>
<i>Ilustración 60 - Estructura física proyectos DicereGram .....</i>	<i>89</i>

<i>Ilustración 61 - Abrir Archive Manager Visual Studio 2017</i> .....	93
<i>Ilustración 62 - Archive Manager Visual Studio 2017</i> .....	94
<i>Ilustración 63 - Distribución aplicación Visual Studio 2017</i> .....	94
<i>Ilustración 64 - Firma aplicación Visual Studio 2017</i> .....	95
<i>Ilustración 65 - Crear firma para aplicación Android</i> .....	95
<i>Ilustración 66 - Selección firma</i> .....	96
<i>Ilustración 67 - Página principal Google Play Console</i> .....	96
<i>Ilustración 68 - Primer formulario crear aplicación</i> .....	97
<i>Ilustración 69 - Ficha Play Store</i> .....	97
<i>Ilustración 70 - Página web DicereGram</i> .....	98
<i>Ilustración 71 - Edición versión aplicación</i> .....	99
<i>Ilustración 72 - Lanzamiento versión aplicación</i> .....	99

# 1. Introducción

---

En este proyecto se plantea el desarrollo de una aplicación móvil Android cuyo objetivo primordial es permitir al colectivo de usuarios invidentes o con discapacidad visual utilizar las funciones de la aplicación móvil Telegram.

La aplicación móvil desarrollada se comunicará con el API de Telegram, y mediante la implementación del reconocimiento y la síntesis de voz en el proyecto, nos permitirá alcanzar el objetivo de ayudar al colectivo invidente a comunicarse con mayor facilidad.

El usuario, mediante el uso de la aplicación móvil, será capaz de enviar un mensaje a un contacto a través de la voz, escuchar un mensaje recibido, buscar un contacto, bloquear o desbloquear un contacto, o realizar ajustes de configuración de la aplicación, entre otras funcionalidades.

Se hará uso de la librería TLSharp<sup>2</sup>, la cual es una librería no oficial del cliente de API de Telegram implementada en lenguaje C#.

En resumen, se plantea desarrollar una aplicación móvil con la que, sin necesidad de interactuar con la pantalla de dispositivo, se permita una comunicación fácil y rápida al colectivo de personas invidentes mediante una aplicación de mensajería móvil.

---

<sup>2</sup> TLSharp. [Sitio web oficial](#).

## 2. Marco teórico o Estado del arte

Una vez definida la idea a desarrollar, se ha realizado una investigación y búsqueda de soluciones análogas y de estudios previos relacionados con el tema planteado.

### 2.1. Aplicaciones móviles

#### 2.1.1. Blind Communicator

La aplicación Blind Communicator<sup>3</sup> Launcher de código libre cuyo objetivo es que personas con discapacidad visual puedan usar smartphones y tablets. La aplicación posee una guía de voz que informa al usuario sobre todo lo que ocurre en el dispositivo (pantalla apagada/encendida, rotación de la pantalla, llamada entrante, etc).

La aplicación le permite al usuario:

- Escuchar música.
- Usar una grabadora de voz.
- Enviar y recibir SMS.
- Recibir y realizar llamadas.
- Crear y eliminar alarmas y contactos.
- Navegar por la Web con el motor de Google como punto de partida.



Ilustración 1 - Logo e interfaz Blind Communicator

#### 2.1.2. Deaf-Blind Communicator

La aplicación Deaf-Blind Communicator<sup>4</sup> brinda la oportunidad a las personas con discapacidades, en particular a ciegos, sordos, personas ciegas y sordas, y a personas con visión o audición limitada, de comunicarse entre ellos y con otras personas.



Ilustración 2 - Logo e interfaz Deaf-Blind Communicator



<sup>3</sup> Blind Communicator. [Sitio web oficial](#). [Play Store](#).

<sup>4</sup> Deaf-Blind Communicator. [Sitio web oficial](#). [Play Store](#).

Características:

- Convertir voz a texto y texto a voz.
- Llamada de teléfono mediante comandos de voz.
- Enviar SMS mediante comandos de voz.
- Lista de palabras / frases comunes para facilitar la entrada rápida.
- Dictado de mensajes entrantes y conversión en secuencias de vibración.
- Botón dirección actual y botón SOS (utilizado en situaciones de emergencia para pedir ayuda).

### 2.1.3. Mobility Launcher

Mobility Launcher<sup>5</sup> es una aplicación gratuita y de código libre cuyo objetivo es que personas con movilidad reducida puedan utilizar smartphones y tablets. La aplicación requiere del uso de un dispositivo USB auxiliar, el cual debe estar conectado para funcionar. Este dispositivo posee un LCD para notificaciones y una serie de botones para utilizar un smartphone o tablet. La aplicación posee una guía de voz que informa al usuario sobre todo lo que pasa en el dispositivo (pantalla apagada/encendida, rotación de la pantalla, llamada entrante, etc.). Con esta aplicación y el dispositivo USB auxiliar conectado, se puede utilizar un smartphone o tablet incluso con la pantalla apagada. Esta aplicación funciona en dispositivos Android 3.2 que tengan habilitada la función de USB Host.



Ilustración 3 - Logo Mobility Launcher

La aplicación le permite al usuario:

- Escuchar música.
- Usar una grabadora de voz.
- Enviar y recibir SMS.
- Recibir y hacer llamadas.
- Crear y eliminar alarmas.
- Crear y eliminar contactos.
- Navegar por la Web con el motor de Google como punto de partida.



Ilustración 4 - Interfaz Mobility Launcher

<sup>5</sup> Mobility Launcher. [Sitio web oficial](#). [Play Store](#).

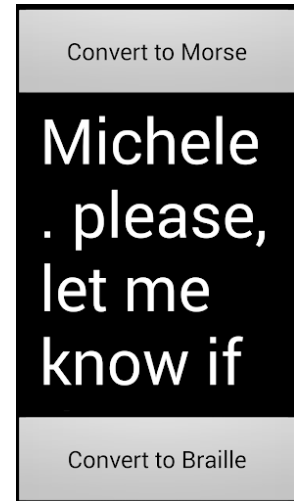
## 2.1.4. Blind SMS Reader 3.2

Blind SMS Reader 3.2<sup>6</sup> es una aplicación diseñada para facilitar la vida a las personas ciegas, deficientes visuales y/o sordas mediante una lectura avanzada del código Morse. También puede leer Braille en dispositivos de pantalla táctil.

Es una aplicación de pago, por lo tanto la versión gratuita tiene la limitación de que solo se lee el 50% del contenido del mensaje.



Ilustración 5 - Logo e interfaz Blind SMS Reader



## 2.1.5. Blind Navigator

La aplicación Blind Navigator<sup>7</sup> tiene 8 principales funciones:

- Teléfono: marcar el número al escuchar la voz.
- Contactos: lee el nombre de los contactos.
- Mensaje: lee los mensajes y también compone mensaje de voz y envía SMS.
- Alarma: alarmas establecidas con voz.
- Calculadora: por voz.
- Identificación de color.
- Navegación GPS guiada por voz.
- Servicios para saber información de batería, intensidad de señal, activación/desactivación de GPS.



Ilustración 6 - Logo Blind Navigator

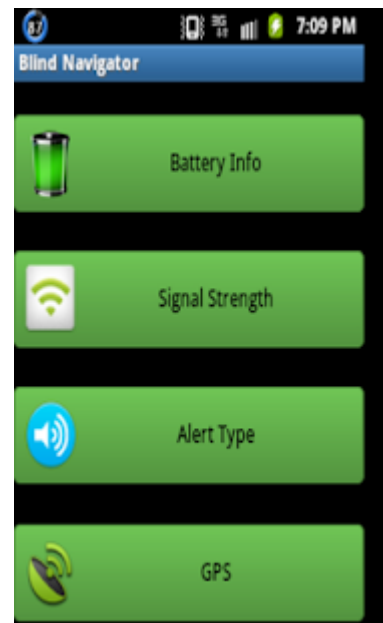


Ilustración 7 - Interfaz Blind Navigator

<sup>6</sup> Blind SMS Reader. [Play Store](#).

<sup>7</sup> Blind Navigator. [Play Store](#).

## 2.1.6. Comaid

Comaid<sup>8</sup> es una aplicación centrada en las personas con discapacidad incluidos ciegos, sordos y mudos. Sin embargo, esta aplicación también es adecuada para el usuario normal.



Ilustración 8 - Logo Comaid

Características:

- Chat en línea para chatear con cualquier persona que utilice esta aplicación en un entorno con Internet.
- Chat offline a través del uso de Bluetooth para comunicarse en un entorno sin Internet.
- Actuar como ayudante personal de comunicación para una conversación en la vida real.
- Traductor universal con la mayor parte del lenguaje cubierto.
- Incluye la función leer en voz alta y el reconocimiento de voz.
- El usuario puede agitar para hablar, agitar de nuevo para enviar el mensaje en el chat en línea y/o en el chat Bluetooth.
- El usuario puede agitar para convertir texto a voz o viceversa.
- El usuario puede agitar para hablar, después de traducir el texto se mostrará y agitar de nuevo para leer en voz alta la palabra traducida.
- El usuario puede cubrir la parte superior de la pantalla del dispositivo con cualquier superficie para leer un mensaje en alto.
- El usuario puede poner la palma de la mano sobre la pantalla para borrar el mensaje.
- El nombre de usuario sólo se tiene que introducir una vez y la aplicación le recordará automáticamente.



Ilustración 9 - Interfaz Comaid

---

<sup>8</sup> Comaid. [Play Store](#).

## 2.2. Estudios previos relacionados

Tras la búsqueda de estudios previos relaciones con el tema propuesto, se han descubierto varios artículos, a destacar uno de ellos, en el que se realiza una investigación con personas con discapacidad visual, y la comparación de la facilidad de uso de dos aplicaciones móvil.

### 2.2.1. Developing Apps for Visually Impaired People: Lessons Learned from Practice [49th Hawaii International Conference on System Sciences 2016]

#### 2.2.1.1. Introducción

Artículo de la 49ª edición de la Conferencia Internacional de Ciencias de Sistemas de Hawaii. En él se realiza una investigación sobre cómo desarrollar aplicaciones para personas con discapacidad visual.

El artículo se basa en la investigación de hábitos de uso de dispositivos móviles por personas con discapacidad visual y en la creación de un prototipo para poder responder la pregunta: ¿Qué tipo de interacción hace a las personas con discapacidad visual más fácil el uso de dispositivos móviles?

#### 2.2.1.2. Entrevistas formativas

Entre las investigaciones realizadas se encuentra una encuesta realizada a 6 participantes con discapacidad visual con edades comprendidas entre 22 y 51 años. Las entrevistas son semiestructuradas, individualmente dirigidas y con una ubicación establecida previamente.

#### 2.2.1.3. Desarrollo y diseño de prototipos

La aplicación prototipo realizada tiene el nombre de All Appointments. Esta aplicación es un reemplazo de la aplicación del calendario nativa.

El objetivo del desarrollo de esta aplicación era crear una aplicación móvil accesible, que fuera capaz de interactuar a través de comandos de voz y del multitouch con personas con discapacidad visual. La interfaz de usuario está diseñada usando un alto contraste entre los colores de fondo y los de primer plano.

A continuación, mostraremos varios de los mockups del prototipo All Appointments realizados por el equipo de investigación de este artículo:





Ilustración 10 - Interfaz All Apointments

#### 2.2.1.4. Estudio de usuario

Para la realización del estudio, los autores del artículo decidieron observar el uso, por parte de los participantes, de la aplicación de calendario nativa y de la aplicación prototipo para conseguir un análisis comparativo entre ellas. Para poder conseguir esto, los usuarios entrevistados recibieron tareas para ser llevadas a cabo con ambas aplicaciones.

Antes de todo, se preguntó a los participantes sobre el recurso de accesibilidad que usan habitualmente, a lo que la mayoría de ellos respondieron que solo suelen usar lectores de pantalla. Además, se les preguntó cómo suelen aprender a usar una nueva aplicación de su dispositivo móvil, a lo que respondieron que intentan explorar la aplicación por ellos mismos, salvo un participante, el cual pregunta a otras personas.

##### a) Procedimiento

Los test fueron realizados primero usando comandos de voz, y luego usando un teclado. Se realizaron los mismos tests tanto para la aplicación de calendario nativa como para el prototipo All Appointments.

##### b) Resultados

Los resultados de los test de los participantes mostraron que la aplicación de calendario nativa era muy difícil de usar y no era nada intuitiva y, además, la ausencia de retroalimentación sonora hacía confundir a los participantes. Los resultados de los test con respecto al prototipo All Appointment revelaron que la aplicación era práctica e intuitiva. Además, manifestaron que la interfaz era mucho más agradable que la de la aplicación de

calendario nativa. Por último, los participantes expusieron que la retroalimentación sonora fue un factor facilitador, el cual mejoraba la experiencia de exploración de la pantalla.

### c) Conclusiones

Las conclusiones obtenidas por los autores del artículo fue principalmente que las interfaces que proporcionan diferentes formas de interacción, como voz, tacto y vibración, facilitan el uso de las aplicaciones a personas con discapacidad visual.

La siguiente tabla muestra un resumen de las acciones aprendidas hasta la etapa actual de la investigación. Se han categorizado utilizando algunas de las heurísticas de usabilidad más conocidas.

Usability heuristics	Lessons Learned
Appropriate feedback	Spoken feedback is essential. Vibration to inform changes.
Aesthetics and minimalist design	Top-to-bottom screen design. Avoid using labels. Avoid hidden components. Keep interaction as simple as possible.
User control and freedom	Natural speech language for voice commands. Provide ways to disable spoken feedback. Provide ways to configure the colors of the interface components.
Recognition rather than recall	Users should be able to easily identify the features and options provided by the interface.

*Ilustración 11 - Conclusiones All Apointments*

## 2.3. Reflexiones

Tras la investigación del marco teórico o estado del arte, podemos concluir que las personas con discapacidad visual tienen una pobre accesibilidad a las aplicaciones móviles de mensajería, debido a que tan sólo existen unas pocas y éstas solo incluyen mensajería a través de SMS.

En el escenario actual, se nos plantea el reto de ayudar a las personas con discapacidad a usar teléfonos inteligentes para mejorar el modo de acceso para reduciendo las dificultades en interacción con una pantalla táctil. Además de contribuir con su inclusión digital, el uso de un teléfono inteligente y sus numerosas aplicaciones también pueden permitir su inclusión en el ambiente laboral.

## 3. Objetivos

---

Tras la realización de las investigaciones expuestas en el apartado anterior, el objetivo principal es desarrollar una aplicación móvil Android con Xamarin que permita a los usuarios, tanto con discapacidad visual como sin ella, poder utilizar las funciones de la aplicación Telegram<sup>9</sup> tan sólo comunicándose con la aplicación a través de la voz.

Uno de los propósitos de la aplicación es dotarla de un lenguaje natural entre el dispositivo y el usuario llevando a cabo para ello conversaciones como si de la vida diaria se tratara.

Haciendo uso de la síntesis de voz de Android, la detección de gestos y el reconocimiento de voz, se pretende proporcionar un uso “eyes free” no sólo para personas con discapacidad visual, pero principalmente desarrollado para mejorar la vida diaria de ese colectivo.

Se ha realizado un enfoque iterativo en lugar de usar una planificación tradicional, realizando versiones mínimas del producto y ampliándolo mediante iteraciones.

---

<sup>9</sup> Telegram. [Sitio web oficial](#).

## 4. Metodología

---

La metodología implementada en el proyecto se basa en una metodología ágil que aúna características de metodologías ágiles, como Scrum<sup>10</sup> y Kanban<sup>11</sup>.

Se ha optado por una metodología con un enfoque iterativo junto con versiones mínimas del producto y ampliaciones sobre ésta en lugar de usar una metodología basada en la planificación tradicional.

### 4.1. Scrum y Kanban

Las características de la metodología ágil Scrum seleccionadas para llevar a cabo el proyecto han sido sprints y reuniones semanales. Se han llevado a cabo un total de 9 sprints para el desarrollo del proyecto junto con reuniones cada tres semanas.

En cada una de las reuniones realizadas durante todo el progreso del proyecto se afrontaban varios temas a tratar:

- Solución de problemas encontrados.
- Demo de la aplicación con lo implementado hasta dicho sprint.
- Retrospectiva.
- Planificación del siguiente sprint.

Las características de la metodología ágil Kanban escogidas para el desarrollo del proyecto han sido la organización del proyecto en tarjetas y la adopción de algunas reglas. Las reglas adoptadas han sido:

- Una tarea no realizada correctamente no debe pasar al siguiente estado hasta que no se corrijan los errores.
- No se debe comenzar una nueva tarea hasta que la tarea anterior no haya sido finalizada.

### 4.2. GitFlow

Para la gestión del flujo de trabajo del proyecto se ha escogido como modelo de flujo una variante de GitFlow<sup>12</sup>.

---

<sup>10</sup> Scrum. [Libro Agile Project Management with Scrum](#).

<sup>11</sup> Kanban. [Tutorial de Atlassian](#).

<sup>12</sup> GitFlow. [Tutorial de Atlassian](#).

El flujo de trabajo de Gitflow define un modelo de ramificación estricto diseñado en torno a la versión del proyecto. Esto proporciona un marco robusto para la gestión de proyectos más grandes.

Gitflow es solo una idea abstracta de un flujo de trabajo de Git. Esto significa que dicta qué tipo de ramas configurar y cómo combinarlas.

En lugar de usar una única rama maestra (master branch), este tipo de modelo de flujo de trabajo se basa en el uso de dos ramas principales para registrar el historial del proyecto. Estas dos ramas principales son:

- Rama **master**: esta rama abarca el código en desarrollo.
- Rama **production**: esta rama contiene el código en producción.

En el proyecto, además de las dos ramas principales, se hace uso de dos subgrupos de ramas:

- TicketX: En estas ramas se implementan nuevas funcionalidades de la aplicación. Cuando estas ramas se finalizan y se realizan test unitarios sobre ellas, se incorporan a la rama master.
- ReleaseX: En estas ramas se incluyen las nuevas funcionalidades incluidas en la rama master para lanzar una nueva versión de la aplicación a la rama de producción.

A continuación, se muestra un diagrama de ejemplo del modelo de flujo GitFlow modificado utilizado en el desarrollo del proyecto:

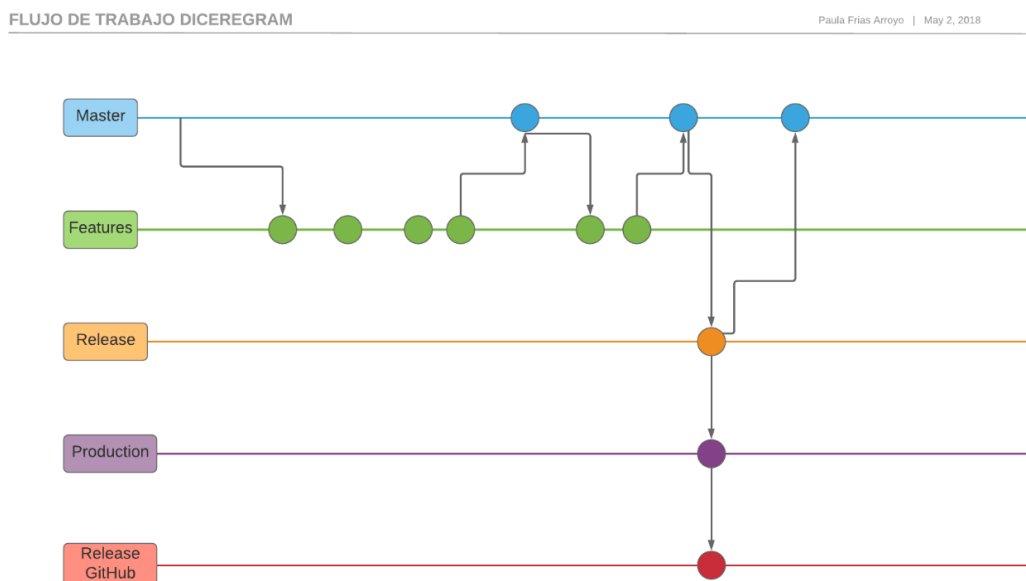


Ilustración 12 - GitFlow

La manera de trabajar que se ha seguido en la implementación de este modelo de flujo de trabajo es la siguiente:

- Desde la rama master se crea una nueva subrama para la implementación de una nueva funcionalidad o el arreglo de algún bug (Ticket).
- Cuando la funcionalidad se haya completado o se haya conseguido solventar el bug (realizando los commits necesarios), se realizará un “pull request” a la rama master.
- Cuando se haya finalizado las funcionalidades mínimas para poder lanzar la aplicación a producción, desde la rama master se creará una nueva subrama dónde se modificará el número de la versión de la aplicación y se realizará un “pull request” sobre la rama production (Release).
- Desde la rama master se realizara un merge con la rama Release para mantener tanto la rama master como la rama production actualizadas.
- Tras esto, ya podemos borrar la rama creada para la realización del Release.

## 4.3. Trello: Tablero

Como herramienta de gestión de proyecto hemos utilizado Trello<sup>13</sup>. Esta herramienta nos permite mantener un control sobre el trabajo realizado en el proyecto durante todo el desarrollo de este.

Al concluir el proyecto se han generado un total de 71 tarjetas de Trello.

### 4.3.1. Listas

Las listas establecidas en el tablero han sido:

- En espera: funcionalidades o bugs del proyecto en espera de ser desarrolladas.
- Seleccionados: funcionalidades o bugs del proyecto seleccionadas para ser desarrolladas en el sprint actual.
- En desarrollo: funcionalidades o bugs del proyecto que están siendo implementados actualmente.
- En desarrollo (done): funcionalidades o bugs del proyecto que ya han sido implementados.

---

<sup>13</sup> Trello. [Sitio web oficial](#).

- Terminados: funcionalidades o bugs del proyecto que han sido implementados y sobre las que se han realizado pruebas funcionalidades sobre el dispositivo físico.

La manera de trabajar ha sido la siguiente:

- Las características de la aplicación (features) se dividen en partes más pequeñas y fáciles de implementar.
  - Estas partes se añaden al backlog (lista de Trello “En espera”).
- Una vez se seleccionan las características a implementar durante el sprint, las tarjetas que correspondan con dichas funcionalidades se pasan a la lista de “Seleccionados” y se añade al nombre de la tarjeta el prefijo “TIC-x”, donde x será el número de ticket (rama) que le corresponde a dicha tarea.
- Cuando se haya creado la rama correspondiente a una tarea, la tarjeta pasará a la lista de “En desarrollo”.
  - Únicamente puede haber una tarjeta en desarrollo, es decir, no se permite trabajar sobre varias tareas de implementación de código en paralelo.
  - Cuando la tarea esté completamente implementada y probada mediante tests unitarios, ésta se pasará a la lista de “En desarrollo (done)”.
  - Una vez se han realizado pruebas de interfaz de usuario sobre un dispositivo físico, las tarjetas se pasarán a la lista de “Terminados” y esto cerrará el ciclo.

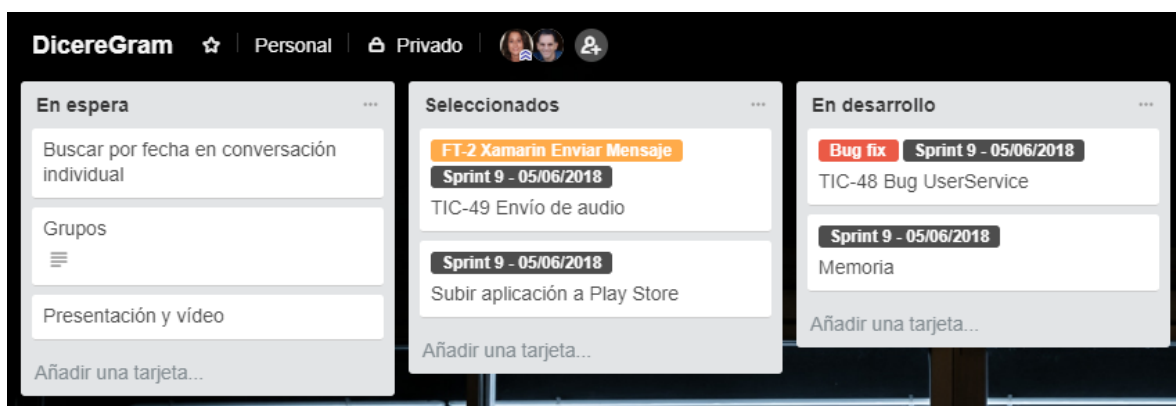


Ilustración 13 - Listas Trello DicereGram

### 4.3.2. Tarjetas

Tal y como puede observarse en la imagen siguiente, las tarjetas de Trello para el proyecto DicereGram se han diseñado de tal forma que contenga tanto la descripción de la tarea a desarrollar como las COS (Conditions Of Satisfaction) de dicha tarea.

También se puede visualizar que las tarjetas que tienen asociada una funcionalidad están compuestas de una checklist con las implementaciones a realizar para que la tarea esté completamente realizada.

Además, las tareas están clasificadas mediante etiquetas que indican a qué feature (característica) pertenece dicha tarea o si es un bug (catalogado como “Bug fix”), y en qué sprint se está realizado.

The screenshot shows a Trello card titled "TIC-3 Enviar mensaje" in the "Terminados" list. It features several key elements:

- Etiquetas:** "Sprint 6 - 09/03/2018" and "FT-2 Xamarin Enviar Mensaje".
- Descripción:** "Como usuario logueado, quiero poder enviar mensajes a cualquiera de mis contactos o grupos a través de la voz."
- COS (Conditions Of Satisfaction):** A list of seven user stories describing the voice message sending process, from user selection to local storage.
- Checklist:** A checklist titled "Enviar Android" with a 100% completion bar and seven items, all of which are checked.
- Acciones:** A sidebar on the right with buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", "Power-Ups" (Google Drive), "Mover", "Copiar", "Seguir", and "Archivar".

Ilustración 14 - Ejemplo tarjeta Trello DicereGram



## 4.4. Control de versiones

El control de versiones del proyecto DicereGram se ha realizado haciendo uso de la herramienta GitHub.

GitHub es una plataforma de desarrollo colaborativo para alojar proyecto Es el control de versiones remoto más conocido. Es compatible con el flujo de trabajo que se ha descrito anteriormente con respecto a la forma de administrar las ramas y los pull requests.

En la recta final del desarrollo del proyecto se han realizado un total de 226 commits repartidos a lo largo del proyecto. A continuación se muestra el gráfico de realización de commits durante la vida del proyecto DicereGram

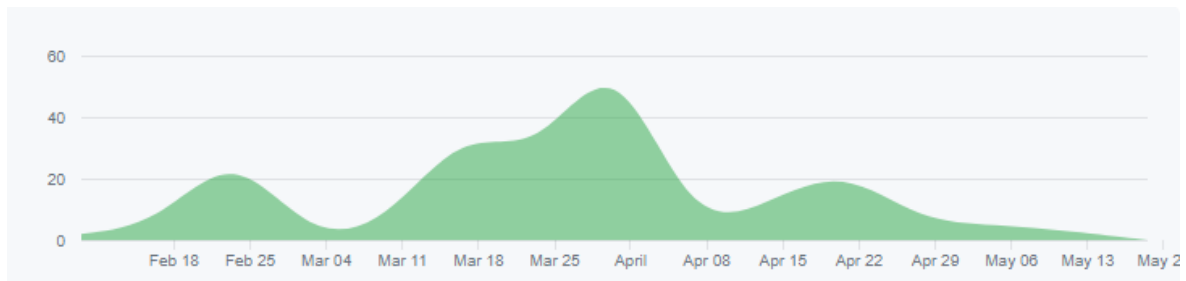


Ilustración 15 - Gráfica 1 commits DicereGram



Ilustración 16 - Gráfica 2 commits DicereGram

En la siguiente gráfica se muestra la red de ramas real del proyecto:

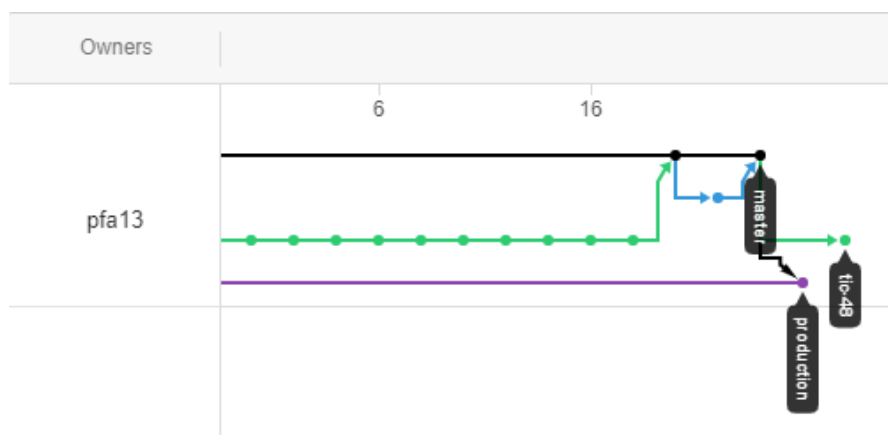


Ilustración 17 - Ramas GitHub

## 4.5. Releases

Se han lanzado un total de 3 releases. Un release abarca un conjunto de sprints que contienen algunas características implementadas.

Los releases del proyecto son los siguientes:

Versión v1.0.0:

- POC (pruebas de concepto):
  - Conexión a la API de Telegram desde .NET
  - Conseguir los contactos de Telegram desde .NET
  - Enviar mensajes con Telegram desde .NET
  - Recibir mensajes de Telegram desde .NET
- Implementación aplicación final:
  - Vista e implementación menú principal
  - Vista e implementación login / registro
  - Vista e implementación enviar mensaje
  - Vista e implementación menú contactos
  - Vista e implementación buscar contacto
  - Vista e implementación contactos bloqueados
  - Implementación actualizar contactos
  - Implementación eliminar contacto
  - Vista e implementación conversaciones generales
  - Vista e implementación conversación individual
  - Implementación borrar conversación con un contacto
  - Vista e implementación menú configuración
  - Implementación aumentar velocidad voz
  - Implementación cambiar nombre de usuario en Telegram
  - Implementación cambio tipo de voz
  - Implementación desactivación voz
- Implementación servicio Android para la recepción de mensajes cada 2 minutos
- (Bug) Implementación control número de teléfono en login
- Incluir logo de la aplicación

- (Bug) Implementación control speech o recognizer al cerrar la aplicación
- Implementación control cuando el usuario no habla, no ha dicho una opción correcta o no se le ha entendido
- Implementación opción de volver atrás o no hacer nada en cada vista.
- (Bug) Implementación control usuario solo diga el nombre estrictamente en todas las vistas
- Implementación control stop service en todas las vistas
- Documento interacción conversaciones
- (Bug) Implementación escribir texto en vista enviar mensaje y cambio colores de cuadros de texto
- Implementación leer mensajes nuevos al entrar a conversaciones general
- Implementación control número de caracteres leídos (máximo 4000)
- (Bug) Implementación control Internet en todas las vistas
- (Bug) Implementación control búsqueda sms code durante dos minutos
- (Bug) Implementación recarga automática de vista al recibir un mensaje nuevo

#### Versión v1.0.1:

- Logout de Telegram y de DicereGram
- Permitir al usuario volver atrás si se equivoca al decir un contacto o éste no existe
- Control de excepciones
- Readme.md
- (Bug) Control cantidad de caracteres pasadas a TTS
- (Bug) Corrección bugs UI en vista conversaciones
- (Bug) Corrección bug al conseguir los usuarios bloqueados
- (Bug) Corrección lectura SMS

## 5. Tecnologías

---

Las tecnologías usadas para el desarrollo del proyecto han sido Xamarin para la implementación de la aplicación Android, SQLite para el almacenamiento en bases de datos, Android TTS para la síntesis de texto a voz, Android Recognition Listener para el reconocimiento de voz, Android Gesture Detector para la detección de gestos en la pantalla y TlSharp para la conexión con el API de Telegram.

### 5.1. Xamarin - Visual Studio 2017

Xamarin<sup>14</sup> es un entorno de desarrollo de aplicaciones que, haciendo uso de código C# para la lógica y de código XAML para las vistas, permite desarrollar aplicaciones nativas para múltiples plataformas (iOS, Android, Mac, Windows Phone, Windows Application, etc).



Ilustración 18 - Logo Xamarin

Las características de Xamarin.Android son:

- Carga de paquetes nativos de Android: Xamarin.Android usa compilación “just-in-time” para una sofisticada optimización del rendimiento de la aplicación en tiempo de ejecución.
- Acceso a cualquier API de Android.
- Llamadas a código java existente desde C#.
- Desarrolla aplicaciones Android Wear.
- Actualizado con Android.



Ilustración 19 - Logo Visual Studio 2017

El IDE utilizado para el desarrollo con Xamarin ha sido Visual Studio 2017<sup>15</sup>.

### 5.2. SQLite

SQLite<sup>16</sup> es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado.



Ilustración 20 - Logo SQLite

---

<sup>14</sup> Xamarin. [Sitio web oficial](#).

<sup>15</sup> Visual Studio 2017. [Sitio web oficial](#).

<sup>16</sup> SQLite. [Sitio web oficial](#).

Las características de SQLite son:

- La base de datos se almacena en un único fichero.
- No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro.

## 5.3. Android TTS

Text-To-Speech (TTS) <sup>17</sup> es un tipo de aplicación de síntesis de voz que se utiliza para crear un sonido hablado del texto introducido.

Android TTS permite la lectura de la información de la pantalla para personas con discapacidad visual o simplemente se puede usar para aumentar la lectura de un mensaje de texto.

Para hacer uso de este paquete se deberá extender e implementar la interfaz `TextToSpeech.OnInitListener`.

A continuación se muestra el código utilizado para hacer uso de Android TTS:

```
TextToSpeech toSpeech = null;
toSpeech = new TextToSpeech(this, this);
```

Tras realizar la creación de un nuevo objeto `TextToSpeech`, Android TTS invoca automáticamente al método `OnInit`. El código mostrado a continuación es un ejemplo de implementación del método.

```
public void OnInit([GeneratedEnum] OperationResult status)
{
    if (status == OperationResult.Success)
    {
        try
        {
            var list = toSpeech.Voices;
            foreach (var item in list)
            {
                if (item.Name == configuracion.TipoVoz)
                {
                    v = item;
                    break;
                }
            }
        }
    }
}
```

---

<sup>17</sup> Android TTS. [Sitio web oficial](#).

```

        if (v != null)
            toSpeech.SetVoice(v);
        toSpeech.SetSpeechRate(configuracion.Velocidad);
        toSpeech.SetLanguage(new Locale("es", "ES"));
        toSpeech.Speak(textToSpeak, QueueMode.Flush, null);
        Thread.Sleep(2000);
    }
    catch (Exception e)
    {
        Toast.MakeText(this, e.Message, ToastLength.Long).Show();
    }
    while (true)
    {
        if (!toSpeech.IsSpeaking)
        {
            Log.Info(LOG_TAG, "he terminado de hablar");
            toSpeech.Stop();
            toSpeech.Shutdown();
            break;
        }
    }
    if (record)
        speechReco.StartListening(intentReco);
    else if (!record && textToSpeak == "Sesión cerrada correctamente")
        this.FinishAffinity();
}
}

```

## 5.4. Android Speech Recognizer

El servicio Android Speech Recognizer<sup>18</sup> permite el acceso al servicio de reconocimiento de voz. Los métodos de esta clase deben invocarse solo desde el hilo principal de la aplicación.

La implementación de esta API es probable que transmita audio a servidores remotos para realizar reconocimiento de voz. Como tal, esta API no está destinada a ser utilizada para el reconocimiento continuo, que consumiría una cantidad significativa de batería y ancho de banda.

<sup>18</sup> Android Speech Recognizer. [Sitio web oficial](#).

Para hacer uso de este paquete se deberá extender e implementar la interfaz `ISpeechRecognizer`.

A continuación se muestra el código utilizado para hacer uso de `Android Speech Recognizer`:

```
SpeechRecognizer speechReco;  
  
Intent intentReco;  
speechReco = SpeechRecognizer.CreateSpeechRecognizer(this.ApplicationContext);  
speechReco.SetRecognitionListener(this);  
intentReco = new Intent(RecognizerIntent.ActionRecognizeSpeech);  
intentReco.PutExtra(RecognizerIntent.ExtraLanguagePreference, "es");  
intentReco.PutExtra(RecognizerIntent.ExtraCallingPackage, this.PackageName);  
intentReco.PutExtra(RecognizerIntent.ExtraLanguageModel,  
RecognizerIntent.LanguageModelWebSearch);  
intentReco.PutExtra(RecognizerIntent.ExtraMaxResults, 1);
```

Una vez se ha utilizado `Android TTS` para comunicar la aplicación con el usuario, comenzaremos a escuchar la respuesta del usuario mediante la siguiente línea de código:

```
speechReco.StartListening(intentReco);
```

Cuando `Android Speech Recognition` detecte que el usuario ha terminado de hablar, se invocará automáticamente al método `OnResults`. A continuación se muestra un fragmento de código de ejemplo de dicho método.

```
public void OnResults(Bundle results) {  
    Log.Info(LOG_TAG, "onResults");  
    IEnumerable<string> matches =  
results.GetStringArrayList(SpeechRecognizer.ResultsRecognition);  
    int countOnResults = 0;  
    count++;  
    record = true;  
    foreach (string result in matches)  
    {  
        countOnResults++;  
        if(countOnResults == 1)  
        {  
            if(count == 1)  
            {  
                if (result.ToLower().Contains("enviar"))  
                {  
                    StopItems();  
                    StartActivity(typeof(Enviar));  
                }  
            }  
        }  
    }  
}
```

```

else if (result.ToLower().Contains("leer")) {
    StopItems();
    StartActivity(typeof(MisChats));
}
else if (result.ToLower().Contains("contactos"))
{
    StopItems();
    StartActivity(typeof(MisContactos));
}
else if (result.ToLower().Contains("configuracion") ||
result.ToLower().Contains("configuración"))
{
    StopItems();
    StartActivity(typeof(Configuracion));
}
else if (result.ToLower().Contains("nada")) { }
else if (result.ToLower().Contains("cerrar sesión") ||
result.ToLower().Contains("cerrar sesion"))
{
    textToSpeak = "¿Está seguro que quiere cerrar la sesión?
Hasta dentro de 2 horas no podrá volverse a loguear con el mismo número de
teléfono.";

    toSpeech = new TextToSpeech(this, this);
}
else
{
    textToSpeak = "Lo siento, no ha dicho una opción válida.
¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión o no hacer
nada?";

    toSpeech = new TextToSpeech(this, this);
}
}
else if(count == 2)
{
    if(result.ToLower().Contains("sí") ||
result.ToLower().Contains("si"))
{
    Logout();
}
else if(result.ToLower().Contains("no"))
{
    count = 0;
    record = true;
    textToSpeak = "¿Qué quiere hacer?. ¿Enviar, leer,
contactos, configuración, cerrar sesión, o no hacer nada?";

```



```

        toSpeech = new TextToSpeech(this, this);
    }
    else
    {
        count = 1;
        record = true;
        textToSpeak = "¿Está seguro que quiere cerrar la
sesión?";

        toSpeech = new TextToSpeech(this, this);
    }
    }
}
}
}
}

```

## 5.5. Android Gesture Detector

El servicio Android Gesture Detector<sup>19</sup> detecta varios gestos y eventos utilizando los MotionEvents suministrados. La devolución de llamada GestureDetector.OnGestureListener notificará a los usuarios cuando se produzca un evento de movimiento en particular. Esta clase solo se debe usar con MotionEvents informados a través del tacto.

Para hacer uso de este paquete se deberá extender e implementar la interfaz GestureDetector.OnGestureListener.

A continuación se muestra el código utilizado para hacer uso de Android Gesture Detector:

```

GestureDetector;
gestureDetector = new GestureDetector(this);

```

Una vez creado el objeto GestureDetector, la aplicación estará siempre a la escucha de cualquier tipo de gestos sobre la pantalla de dispositivo. En el momento en el que se detecte la realización de swipe hacia cualquier de las cuatro direcciones, se realizará acciones de navegación hacia atrás o hacia otra vista. A continuación se muestra un código de ejemplo utilizado en la aplicación DicereGram para controlar los swipe izquierdo, derecho, arriba y abajo:

<sup>19</sup> Android Gesture Detector. [Sitio web oficial](#).

```

public bool OnFling(MotionEvent event1, MotionEvent event2, float velocityX, float
velocityY)
{
    bool result = false;
    try
    {
        float diffY = event2.GetY() - event1.GetY();
        float diffX = event2.GetX() - event1.GetX();

        if (System.Math.Abs(diffX) > System.Math.Abs(diffY))
        {
            if (System.Math.Abs(diffX) > _swipeThresold &&
System.Math.Abs(velocityX) > _swipeVelocityThresold)
            {
                if (diffX > 0) // Right
                {
                    StopItems();
                    StartActivity(typeof(MisChats));
                }
                else // Left
                {
                    StopItems();
                    StartActivity(typeof(Enviar));
                }
            }
        }
        else if (System.Math.Abs(diffY) > _swipeThresold &&
System.Math.Abs(velocityY) > _swipeVelocityThresold)
        {
            if (diffY > 0) // Bottom
            {
                StopItems();
                StartActivity(typeof(Configuracion));
            }
            else // Top
            {
                StopItems();
                StartActivity(typeof(MisContactos));
            }
        }
        result = true;
    }
    catch (Exception exception)
    {
        Toast.MakeText(this, exception.Message, ToastLength.Long).Show();
    }
}

```

```
    }  
    return result;  
}
```

## 5.6. TLSharp

TLSharp<sup>20</sup> es una biblioteca no oficial de cliente de Telegram implementada en C#.

El código siguiente muestra un ejemplo de uso de dicha librería.

```
private int api_id = 190786;  
private string api_hash = "b7e8ca8d987b1fd6f7e068c558a7cd8f";  
public TLClient Connect()  
{  
    try  
    {  
        FakeSessionStore session = new FakeSessionStore();  
        TLClient client = new TLClient(api_id, api_hash, session, "session");  
        var t = Task.Run(() => client.ConnectAsync());  
        t.Wait();  
        return client;  
    }  
    catch(Exception ex)  
    {  
        return null;  
    }  
}
```

## 5.7. Telegram

Telegram es una aplicación de mensajería. El servicio está enfocado en el envío y recibo de mensajes de texto y multimedia y es administrada por una organización sin ánimo de lucro.

Telegram ofrecen dos tipos de APIs<sup>21</sup> para desarrolladores. El Bot API le permite crear fácilmente programas que usan mensajes de Telegram para una interfaz. Mientras que el API de Telegram le permite crear sus propios clientes personalizados de Telegram. Ambas APIs son completamente gratuitas.

---

<sup>20</sup> TLSharp. [Sitio web oficial \(GitHub\)](#).

<sup>21</sup> APIs Telegram. [Sitio web oficial](#).

# 6. Arquitectura e implementación

---

## 6.1. Primeras aproximaciones

Durante el desarrollo del proyecto, se realizaron varios prototipos de la aplicación. Estos prototipos se explican a continuación con más detalle.

### 6.1.1. Kotlin

#### 6.1.1.1. Introducción

En las primeras iteraciones del desarrollo del proyecto, éste se centró en el desarrollo del proyecto mediante la implementación de una aplicación Android haciendo uso del IDE Android Studio y empleando el lenguaje Kotlin<sup>22</sup>.

Se decantó por este lenguaje, ya que era un lenguaje de nueva creación y suponía un reto más en el desarrollo del proyecto.

Kotlin es un lenguaje de programación de tipado estático para aplicaciones multiplataforma modernas. Es 100% interoperable con Java y Android. Permite desarrollar aplicaciones Android, web (JavaScript), del lado del servidor y nativas.

Las características de Kotlin son las siguientes:

- Conciso: reduce drásticamente la cantidad de código repetitivo.
- Seguro: evita clases de errores como excepciones de punteros a nulos
- Interoperable: aprovecha las librerías existentes para JVM, Android y el navegador
- Herramienta amigable: permite elegir cualquier IDE de Java o compilar desde la línea de comandos.

#### 6.1.1.2. Implementación

La implementación de esta aproximación se estuvo desarrollando durante los meses de septiembre, octubre y noviembre de 2017.

Durante este periodo, se registraron las tareas que se fueron realizando en Trello:

---

<sup>22</sup> Kotlin. [Sitio web oficial](#).



Ilustración 21 - Lista Terminados Trello proyecto Kotlin

El repositorio de GitHub con el proyecto en Kotlin de esta primera aproximación se encuentra en el siguiente enlace: <https://github.com/pfa13/DemoKotlin>.

El gráfico de commits que muestra el trabajo realizado durante esta aproximación es el siguiente:

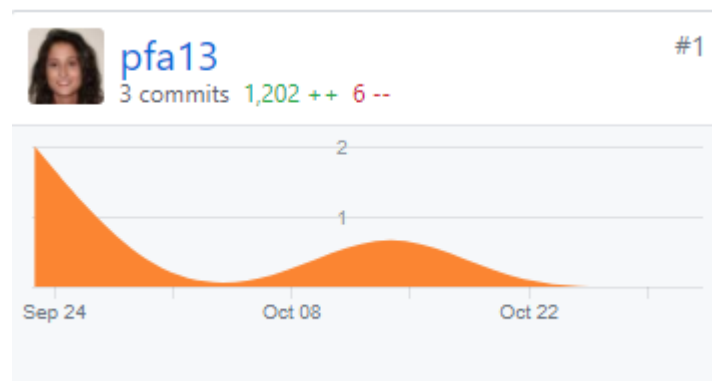


Ilustración 22 - Gráfica commits proyecto Kotlin

### 6.1.1.3. Conclusión

No se continuó con la implementación del proyecto en Kotlin debido a los numerosos problemas que surgieron para poder utilizar el API de Telegram a través de dicho lenguaje:

- El API de Telegram está algo desfasado puesto que lleva más de 4 años sin actualizarse.
- El API de Telegram se basa en el uso de peticiones RPC y haciendo pruebas con Java y Android, el API no realizaba el comportamiento esperado.
- Tras investigar la existencia de librerías de cliente del API de Telegram para Java, Android o Kotlin, no se encontró ninguna librería ni siquiera no oficial para poder realizar el proyecto.

## 6.1.2. API Rest en C#

### 6.1.2.1. Introducción

La siguiente aproximación en la que se estuvo trabajando fue una arquitectura API Rest en lenguaje C# desarrollada en el IDE de Visual Studio 2017 junto con el desarrollo de la aplicación móvil con Kotlin comentada en el apartado anterior.

Se continuó con esta aproximación debido al descubrimiento de una librería no oficial de cliente del API de Telegram desarrollado en C#, TLSharp, la cual funcionaba a la perfección.

### 6.1.2.2. Implementación

Por un lado, el API Rest implementado en C# se encargaría de almacenar las sesiones de los diferentes usuarios de la aplicación, junto con sus datos de acceso y mensajes. Esto permitiría centralizar el control de las sesiones y el control de las llamadas al API de Telegram y poder hacer el proyecto más robusto, haciendo uso de una arquitectura de estas características.

Por otro lado, la aplicación Kotlin realizaría las llamadas al API Rest implementado en el proyecto para poder realizar las funcionalidades de una aplicación de mensajería correctamente.

La implementación de esta aproximación se estuvo desarrollando durante los meses de diciembre de 2017 y enero de 2018.

Durante este periodo, se registraron las tareas que se fueron realizando en Trello:

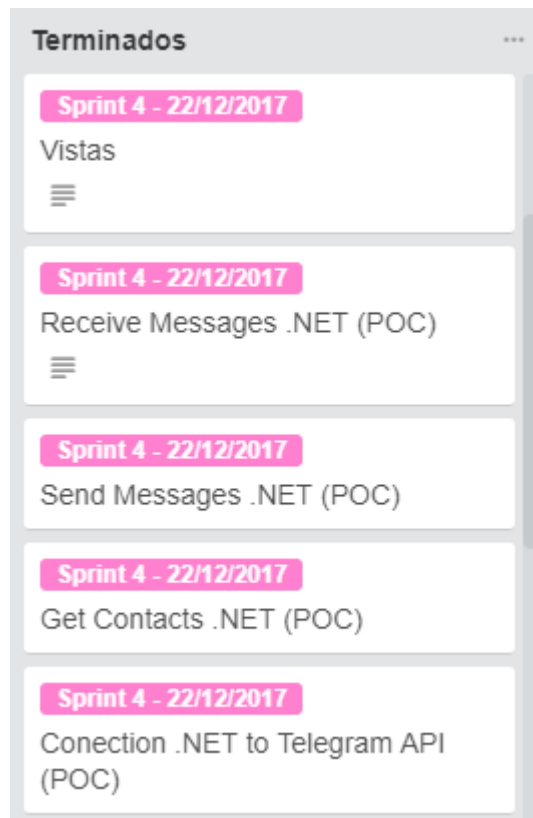


Ilustración 23 - Lista Terminados Trello proyecto API Rest

El repositorio de GitHub con la implementación del proyecto del API Rest en C# de esta primera aproximación se encuentra en el siguiente enlace: <https://github.com/pfa13/DicereGramAPI>.

El gráfico de commits que muestra el trabajo realizado durante esta aproximación es el siguiente:

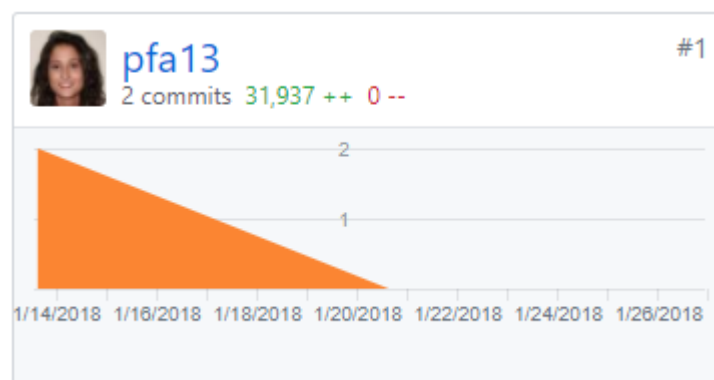


Ilustración 24 - Gráfica commits proyecto API Rest

### 6.1.2.3. Conclusión

No se continuó con la implementación del proyecto en Kotlin con un API Rest en C# debido a los numerosos problemas que surgieron a la hora de controlar las sesiones en el servidor de la API:

- El API de Telegram está pensado para que una y solo una sesión sea creada en cada dispositivo.
- Cuando se creaban varias sesiones en el servidor de la API, Telegram baneaba a los usuarios de 4 a 24 horas.

### 6.1.3. APIs de reconocimiento de voz

#### 6.1.3.1. DialogFlow

DialogFlow<sup>23</sup> (anteriormente denominado api.ai) es un desarrollo de Google de tecnologías de interacción humano-ordenador basadas en conversaciones en lenguaje natural.



*Ilustración 25 - Logo Api.ai*

Permite desarrollar experiencias conversacionales naturales y ricas.

Ofrece al usuario distintas formas de interactuar con una aplicación mediante la construcción de interfaces conversacionales de texto y voz, basadas en inteligencia artificial.

Sus características son:

- Desarrollado por el machine learning de Google.
- Construido en la arquitectura de Google (Google Cloud Platform): lo que permite una mayor escalabilidad de usuarios.
- Optimizado para el asistente de Google



*Ilustración 26 - Logo DialogFlow*

Los precios de DialogFlow son:

---

<sup>23</sup> DialogFlow. [Sitio web oficial.](#)



	Free Standard Edition	Pay as you go Enterprise Edition
Text queries	Free usage with unlimited requests	Unlimited requests at \$0.002 per request
Voice interaction	Free usage up to 1,000 requests per day with a maximum of 15,000 requests per month	Unlimited Cloud Speech-to-Text requests at \$0.0065 per 15 seconds ( <a href="#">see details</a> )
Default quota for text queries	3 queries per second (averaged over a minute)	10 queries per second (averaged over a minute)
Service level agreement	None	<a href="#">Dialogflow Enterprise Edition Service Level Agreement (SLA)</a>
Support	Community and email support	Eligible for <a href="#">Google Cloud Support</a>
Terms of Service	<a href="#">Dialogflow ToS</a>	<a href="#">Google Cloud Platform ToS</a>
Choose a plan →	Ideal for small to medium businesses or those who want to experiment with Dialogflow <a href="#">START NOW</a>	Ideal for businesses that need to easily scale to support changes in demand from their users <a href="#">LEARN MORE</a>

Ilustración 27 - Tabla de precios DialogFlow

### 6.1.3.2. WIT.AI

Wit.ai<sup>24</sup> facilita a los desarrolladores la capacidad de crear aplicaciones a las que puedes hablar o enviarles mensajes de texto.

Ofrece a los desarrolladores una plataforma de lenguaje natural abierta y extensible.



Ilustración 28 - Logo Wit.ai

Wit.ai aprende el lenguaje humano en cada interacción y aprovecha la comunidad (lo que se aprende se comparte entre todos los desarrolladores).

Sus características son:

- Bots: Facilidad para la creación de bots basados en texto o voz.
- Aplicaciones móviles: permite crear una interfaz basada en voz para su aplicación.
- Domótica: permite la automatización del hogar.
- Dispositivos portátiles: permite crear una interfaz basada en voz para estos dispositivos.
- Robots: permite a tu bot entender a los humanos y, además, te permite descubrir necesidades.

Wit.ai es gratuito e incluye uso comercial.

<sup>24</sup> Wit.ai. [Sitio web oficial](#).

### 6.1.3.3. Google Speech

La API Speech<sup>25</sup> de Google Cloud posibilita que los desarrolladores conviertan audio en texto aplicando potentes modelos de redes neuronales en una API fácil de usar. La API reconoce más de 80 idiomas.

Permite transcribir el texto que los usuarios dictan al micrófono de una aplicación, habilitar el control por voz o transcribir archivos de audio, entre muchas otras funciones.

La precisión de la API Speech mejora con el tiempo debido a que está basado en algoritmos de redes neuronales basados en el aprendizaje.

Sus características son:

- Reconocimiento automático de voz.
- Vocabulario internacional.
- Reconocimiento en directo.
- Palabras clave.
- Compatibilidad con audio grabado y en tiempo real.
- Tratamiento del ruido.
- Filtrado de contenido inapropiado.
- API integrada.

Los precios de la API Speech de Google son:

USO MENSUAL	PRECIO POR CADA 15 SEGUNDOS*
Hasta 60 minutos	Gratuito
61 - 1.000.000 de minutos**	0,006 \$

Ilustración 29 - Tabla de precios API Google Speech

### 6.1.3.4. LUIS (Language Understanding Intelligent Service)

Luis<sup>26</sup> es un servicio basado en el aprendizaje automático para construir lenguaje natural en aplicaciones, bots y dispositivos IoT.

---

<sup>25</sup> Google Speech. [Sitio web oficial.](#)

<sup>26</sup> Luis. [Sitio web oficial europeo.](#)

Language Understanding Intelligent Service (LUIS) permite a su aplicación comprender lo que una persona quiere con sus propias palabras. LUIS utiliza el aprendizaje automático para permitir a los desarrolladores crear aplicaciones que puedan recibir las entradas de los usuarios en un lenguaje natural y extraer su significado. Una aplicación cliente que conversa con el usuario puede pasar la entrada del usuario a una aplicación LUIS y recibir información relevante y detallada.

Sus características son:

- Añadir lenguaje natural a tus aplicaciones.
- Crear rápidamente una solución de idioma personalizada.

Los precios de LUIS son:

TIER	FEATURES	UNIT	PRICE
LUIS API—Free		Transactions	10,000 transactions free per month
LUIS API—Basic	Up to 50 transactions per second	Transactions	€1.265 per 1,000 transactions

*Ilustración 30 - Tabla de precios LUIS*

### 6.1.3.5. Conclusiones

No se ha optado por el uso de ninguna de las APIs de reconocimiento de voz anteriores debido a que esto suponía tener que realizar una llamada más a una API externa. Se ha optado por realizar el reconocimiento de la aplicación a través del paquete nativo de Android Speech Recognizer, del que se hablará un poco más profundamente en capítulos siguientes.

## 6.2. Diagrama Entidad-Relación

La imagen siguiente muestra el diagrama entidad-relación de la base de datos implementada localmente mediante SQLite en el dispositivo del usuario:

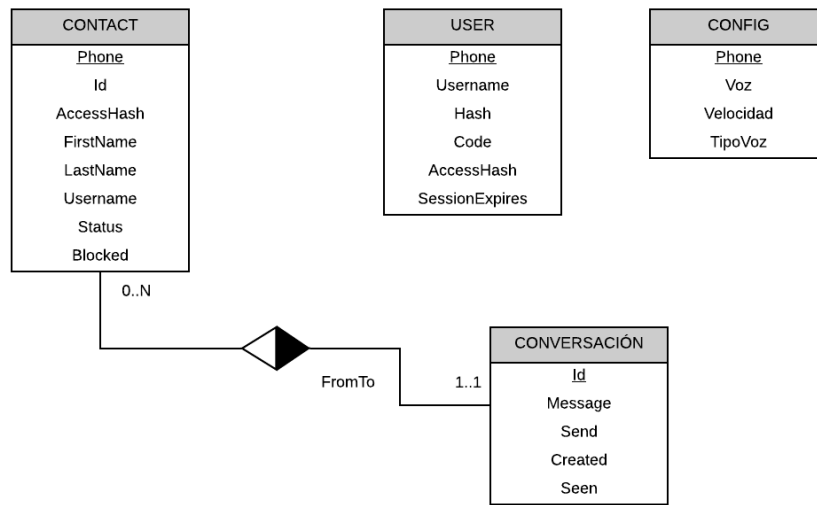


Ilustración 31 - Diagrama Entidad Relación DicereGram

### 6.3. Implementación y control de conversaciones

El diagrama siguiente muestra una visión global de la implementación y arquitectura realizada.

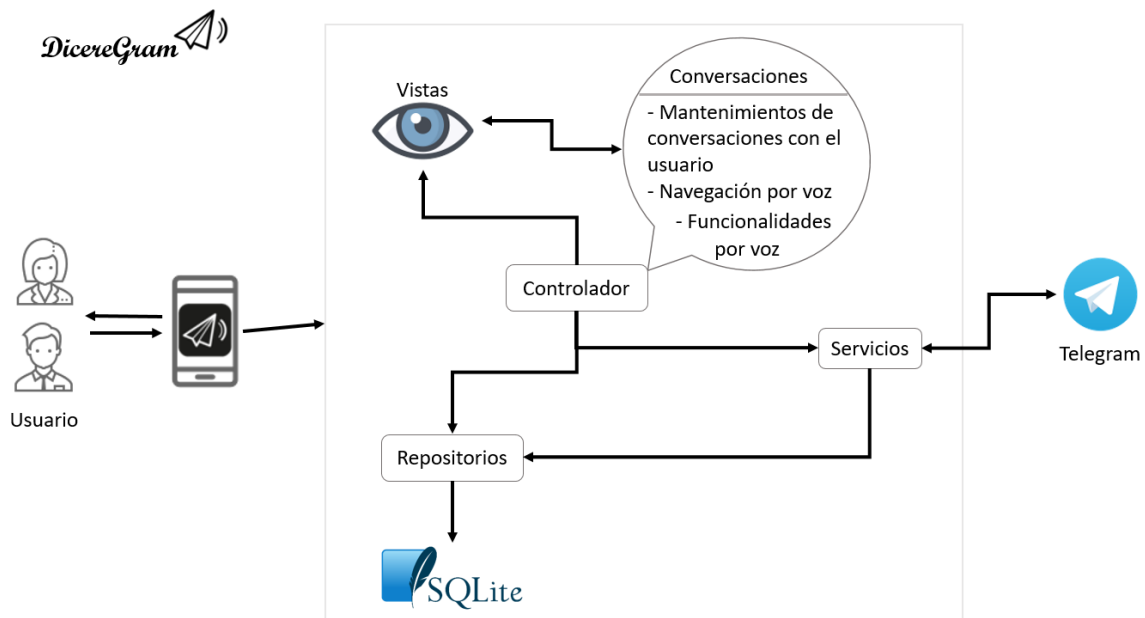


Ilustración 32 - Diagrama arquitectura DicereGram

La arquitectura utilizada es la arquitectura MVC<sup>27</sup> (Modelo, Vista, Controlador) extendida. Se ha añadido la capa de servicios a dicho modelo para realizar las peticiones hacia Telegram. Se ha optado por una arquitectura de estas características debido a que cumple perfectamente con la estructura necesaria para el correcto desarrollo del proyecto DicereGram.

Las capas de la arquitectura MVCS comprenden:

- Modelo:
  - Esta capa es la encargada de efectuar el acceso a los datos a través de la conexión con la base de datos SQLite.
  - Se utiliza el patrón DAO (Data Access Object), separando completamente la lógica de negocio de la lógica de acceso a datos.
  - También se han implementado BO (Business Object) para realizar la organización del código en unidades lógicas.
- Vista:
  - Esta capa se encarga de mostrar la información a través de la interfaz al usuario, y mediante acciones asociadas a la vista, realizar peticiones al sistema.
  - Cada vista tiene asociada un controlador con el que interactúa.
- Controlador:
  - Esta capa es la encargada de recibir las peticiones procedentes de la vista o de un servicio en segundo plano y de generar las respuestas o acciones en respuesta a ellas.
  - Utiliza la capa servicio para realizar las peticiones necesarias al API de Telegram.
  - Además, utiliza la capa de repositorios (capa Modelo) para acceder a la lógica de datos de la aplicación, en los casos en los que, previamente, no es necesario realizar ninguna petición al API de Telegram, mediante llamadas a la capa de Servicio.

---

<sup>27</sup> MVC. [Explicación](#).

- Servicio:
  - La capa de Servicio contiene toda la lógica de negocio para la manipulación de los datos que se obtienen en la capa Modelo. De este modo, separamos la lógica de negocio tanto del controlador como del modelo.
  - Esta capa es la encargada de realizar peticiones al API de Telegram y, además, interactuar con la capa de acceso a datos en los casos necesarios, como por ejemplo en la petición de envío de un mensaje de texto en el que se conectará con la API de Telegram y, también, se almacenará el mensaje en la base de datos local.

## 6.4. Interfaces

Se han implementado interfaces para cada clase, tanto de servicio como de repositorio, implementada en el proyecto. Esto se ha hecho así planificando una posible futura inyección de dependencias en el proyecto, y llegado el caso, la implementación sería muy más rápida y sencilla.

## 6.5. Servicios

La capa de servicio implementada en el proyecto es la única encargada de realizar peticiones a la API de Telegram mediante la librería de cliente de API de Telegram, TLSharp.

La imagen siguiente muestra los servicios desarrollados y todos los métodos implementados en ellos:

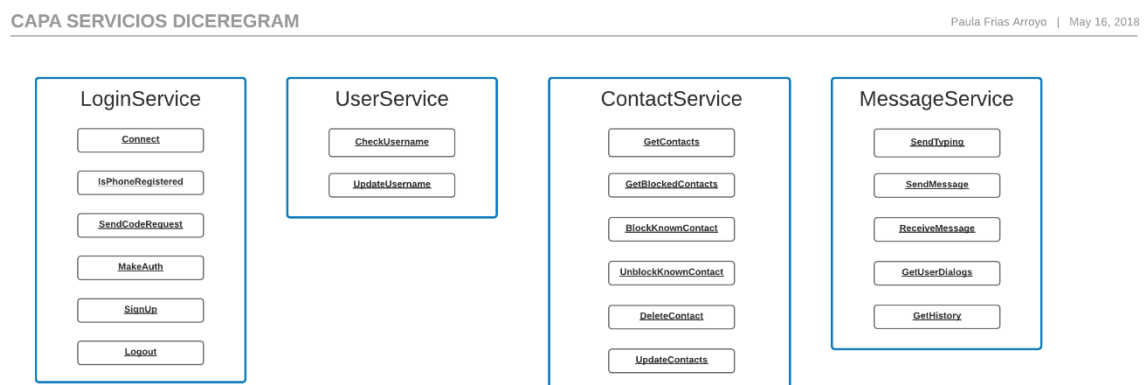


Ilustración 33 - Diagrama capa servicios DicereGram

A continuación se realizará un pequeño resumen de todos los métodos de la API de Telegram utilizados en cada uno de los servicios mostrados en la imagen anterior:

- LoginService:

Método Telegram	Método TLSharp	Descripción
	Connect	Establece la conexión con Telegram
auth.checkPhone	IsPhoneRegisteredAsync	Comprueba si el número de teléfono pasado como parámetro está registrado en Telegram.
auth.sendCode	SendCodeRequestAsync	Envía un mensaje de texto con el código de confirmación requerido para el registro, al número de teléfono pasado como parámetro.
auth.signIn	MakeAuthAsync	Autoriza a un usuario en el sistema basado en el número de teléfono validado pasado como parámetro.
auth.signUp	SignUpAsync	Registra un número de teléfono validado pasado como parámetro
auth.logOut	TLRequestLogout	Desautoriza a un usuario en el sistema.

- UserService:

Método Telegram	Método TLSharp	Descripción
account.checkUsername	TLRequestCheckUsername	Valida el nombre de usuario pasado como parámetro y chequea su disponibilidad.
account.updateUsername	TLRequestUpdateUsername	Actualiza el nombre de usuario del usuario actual.

- ContactService:

Método Telegram	Método TLSharp	Descripción
-----------------	----------------	-------------

contacts.getContacts	GetContactsAsync	Devuelve una lista con los usuarios de los contactos actuales.
contacts.getBlocked	TLRequestGetBlocked	Devuelve la lista de los usuarios bloqueados.
contacts.block	TLRequestBlock	Añade un usuario a la lista de usuarios bloqueados.
contacts.unblock	TLRequestUnblock	Borra un usuario de la lista de usuarios bloqueados.
contacts.deleteContact	TLRequestDeleteContact	Borra un único contacto de la lista de contactos.

- MessageService:

Método Telegram	Método TLSharp	Descripción
message.setTyping	SendTypingAsync	Transmite el evento actual de "escribiendo" del usuario al contacto o grupo de conversación
message.sendMessage	SendMessageAsync	Envía un mensaje de texto a un contacto o grupo de conversación.
message.getDialogs	GetUserDialogsAsync	Devuelve la lista de conversaciones actual del usuario.
message.getHistory	GetHistoryAsync	Devuelve los mensajes del historial de una única conversación.

Para ejemplificar los métodos descritos en las tablas anteriores, se muestran los trozos de código siguientes:

```
public TLClient Connect()
{
    try
    {
        FakeSessionStore session = new FakeSessionStore();
        TLClient client = new TLClient(api_id, api_hash, session,
"session");
    }
}
```



```

        var t = Task.Run(() => client.ConnectAsync());
        t.Wait();
        return client;
    }
    catch (Exception ex)
    {
        throw;
    }
}

```

```

public int ReceiveMessages(TLClient client, MessageRepository
messageRepository)
{
    TLInputPeerUser target = null;
    bool hayNuevos = false;
    int messageCount = 0;
    int firstMessage = 0;
    try
    {
        var dialogs = GetUserDialogs(client);
        IEnumerable<TLDialog> listDialogs =
dialogs.Dialogs.ToList().Where(x => (x.UnreadCount > 0));

        foreach (var dia in listDialogs)
        {
            hayNuevos = true;
            if (dia.Peer is TLPeerUser)
            {
                var peer = dia.Peer as TLPeerUser;
                var chat =
dialogs.Users.ToList().OfType<TLUser>().First(x => x.Id == peer.UserId);
                target = new TLInputPeerUser { UserId = chat.Id,
AccessHash = (long)chat.AccessHash };
                var hist = GetHistory(client, target, dia);
                if (hist is TLMessagesSlice)
                {
                    var h = hist as TLMessagesSlice;
                    var history = h.Messages.ToList();
                    for (var i = 0; i < history.Count; i++)
                    {
                        var mens = history[i] as TLMessage;
                        if (i == 0)
                            firstMessage = mens.Id;
                        if (!mens.Out)
                        {
                            Chat c = new Chat

```



```

        var resultado = affectedMessages.Result;
    }
}
return messageCount;
}
catch (Exception ex)
{
    return -1;
}
}

```

## 6.6. Repositorios

La capa de repositorio implementada en el proyecto es la única encargada de realizar operaciones sobre la base de datos local.

En las tablas siguientes se describen los métodos creados en cada uno de los repositorios del proyecto

- SQLiteRepository:

Método	Descripción
<b>GetPath</b>	Devuelve la ruta de la base de datos local del usuario.
<b>GetConnection</b>	Devuelve una nueva conexión a la base de datos local del usuario.
<b>GetConnectionAsync</b>	Devuelve una nueva conexión asíncrona a la base de datos local del usuario.
<b>CreateDatabase</b>	Crear la base de datos local junto con toda la estructura de entidades.
<b>DeleteDatabase</b>	Borra todos los registro de todas las entidades de la base de datos local

- UserRepository:

Método	Descripción
<b>GetUser</b>	Devuelve la única entrada de datos de la entidad User.

<b>InsertUser</b>	Inserta o reemplaza el usuario dado de alta en la base de datos.
<b>UpdateUser</b>	Actualiza el único usuario almacenado en la entidad User de la base de datos.

- ContactRepository:

<b>Método</b>	<b>Descripción</b>
<b>InsertContact</b>	Inserta o reemplaza un contacto en la base de datos local.
<b>GetContacts</b>	Devuelve una lista con la estructura completa de datos de todos los contactos almacenados en la base de datos ordenados por nombre.
<b>GetContactsByName</b>	Devuelve una lista con los datos de los contactos almacenados en la base de datos cuyo nombre (firstname lastname) esté contenido en el nombre pasado como parámetro.
<b>GetContactByName</b>	Devuelve una lista con los datos del contacto almacenado en la base de datos cuyo nombre (firstname lastname) sea igual al nombre pasado como parámetro.
<b>GetContactsName</b>	Devuelve una lista con el nombre de los contactos almacenados en la base de datos.
<b>GetContactsStatus</b>	Devuelve una lista con el estado actual de los contactos almacenados en la base de datos.
<b>UpdateContact</b>	Actualiza un contacto.
<b>GetBlocked</b>	Devuelve la lista con los contactos que están bloqueados.
<b>GetBlockedName</b>	Devuelve una lista con los nombres de los contactos que están bloqueados.
<b>DeleteContact</b>	Borra un contacto.

<b>DeleteContacts</b>	Borra todos los contactos.
<b>GetContactById</b>	Devuelve los datos de un contacto cuyo identificador coincida con el identificador pasado como parámetro.
<b>GetContactByPhone</b>	Devuelve los datos de un contacto cuyo número de teléfono coincida con el número de teléfono pasado como parámetro.
<b>GetContactsByNameWithChat</b>	Devuelve una lista con los contactos con los que el usuario mantiene una conversación y cuyo nombre de contacto coincida con el nombre pasado como parámetro.

- MessageRepository:

<b>Método</b>	<b>Descripción</b>
<b>GetMessages</b>	Devuelve una lista con los mensajes actuales.
<b>GetMessagesByPhoneOrdered</b>	Devuelve una lista con los mensajes del contacto cuyo número de teléfono coincida con el número de teléfono pasado como parámetro y ordenado de manera descendente por fecha.
<b>GetMessagesByPhone</b>	Devuelve una lista con los mensajes del contacto cuyo número de teléfono coincida con el número de teléfono pasado como parámetro.
<b>GetMessagesOrdered</b>	Devuelve una lista con el último mensaje de cada contacto ordenado a través de la fecha de manera descendente.
<b>GetMessagesNotReaded</b>	Devuelve una lista con los mensajes no leídos.

<b>GetMessagesNotReadedOrderedByContact</b>	Devuelve una lista con los mensajes no leídos ordenado a través de la fecha y número de teléfono del contacto.
<b>CountMessagesNotReaded</b>	Devuelve una lista con la cantidad de mensajes no leídos de cada contacto.
<b>InsertMessage</b>	Inserta un mensaje nuevo en la base de datos.
<b>CheckExistFromContact</b>	Comprueba si existe una conversación con un contacto cuyo número de teléfono coincida con el número de teléfono pasado como parámetro.
<b>GetMessagesByPhoneWithoutSeen</b>	Devuelve una lista con los mensajes no vistos del contacto cuyo número de teléfono coincide con el número de teléfono pasado como parámetro.
<b>GetMessagesByPhoneWithoutSeen (paginado)</b>	Devuelve una lista paginada con los mensajes no vistos del contacto cuyo número de teléfono coincide con el número de teléfono pasado como parámetro.
<b>GetMessagesByPhoneAndMessage</b>	Devuelve una lista con los mensajes entrantes y salientes posteriores al mensaje que contiene la cadena buscada pasada como parámetro del contacto cuyo número de teléfono coincide con el número de teléfono pasado como parámetro.
<b>GetMessagesByPhoneAndMessage (paginado)</b>	Devuelve una lista paginada con los mensajes entrantes y salientes posteriores al mensaje que contiene la cadena buscada pasada como parámetro del

	contacto cuyo número de teléfono coincide con el número de teléfono pasado como parámetro.
<b>MarkMessagesAsRead</b>	Actualiza los mensajes no leídos del contacto pasado como parámetro y los marca como leídos.
<b>DeleteChat</b>	Borra la conversación con el contacto pasado como parámetro.

- ConfigRepository:

Método	Descripción
<b>GetConfig</b>	Recupera la configuración almacenada en la base de datos local.
<b>InsertConfig</b>	Inserta una nueva entrada de datos en la entidad Configuración de la base de datos.
<b>UpdateConfig</b>	Inserta los datos almacenados para la entidad Configuración de la base de datos.

Con el objetivo ejemplarizar los métodos descritos en las tablas anteriores, se muestran los trozos de código siguientes:

```
public List<string> GetContactsStatus()
{
    try
    {
        using (var connection = _sqlite.GetConnection())
        {
            return GetContacts().Select(x => x.FirstName + " " +
x.LastName + " - " + x.Status).ToList();
        }
    }
    catch (SQLiteException ex)
    {
        return null;
    }
}
```

```

public List<Chat> GetMessagesOrdered()
{
    try
    {
        using (var connection = _sqlite.GetConnection())
        {
            return connection.Query<Chat>(@"SELECT Id, FromTo,
Send, Mensaje, max(Created) as Created, Seen
                                FROM Chat
                                GROUP BY FromTo
                                ORDER BY Created
desc");
        }
    }
    catch (SQLiteException ex)
    {
        throw;
    }
}

```

```

public List<Chat> GetMessagesByPhoneAndMessage(string phone, string
message, int position)
{
    try
    {
        using (var connection = _sqlite.GetConnection())
        {
            var list = connection.Query<Chat>(@"SELECT *
                                FROM Chat
                                WHERE FromTo =
'" + phone + "'
                                AND Created
>= (SELECT Created FROM Chat WHERE Mensaje LIKE '%" + message + "' LIMIT
1) ORDER BY Created " +
                                "LIMIT 10
OFFSET " + position.ToString());
            return list;
        }
    }
    catch (SQLiteException ex)
    {
        throw;
    }
}

```



## 6.7. Servicios en segundo plano

Los servicios en segundo plano realizados son los encargados de mantener el servicio en activo en el dispositivo Android, con el objetivo de realizar correctamente la recepción de mensajes.

Los servicios implementados son los siguientes:

- **ReceiveService:** Este servicio es el servicio en segundo plano por excelencia. Si este servicio es parado, vuelve a levantarse a través del servicio de broadcast.
  - Está compuesto por un timer, gracias al que invocamos al BroadcastReceiver `GetMessages`, cada 2 minutos.
  - El método en los controladores, dónde se arranca el servicio si no está arrancado es el siguiente:

```
service = new Intent(this, typeof(ReceiveService));
if (!IsMyServiceRunning(service))
    StartService(service);
```

- **GetMessages:** Clase que extiende de `BroadcastReceiver`<sup>28</sup> e invoca al método de servicio `ReceiveMessages`, con el que recibe los mensajes entrantes del usuario, y en caso de que existan mensajes, notifica al usuario mediante la creación de una notificación.
- **ReceiveMessagesService:** Clase que extiende de `BroadcastReceiver` y es la encargada de volver a levantar el servicio una vez ha sido parado por el cierre de la aplicación `DicereGram`, por parada del servicio por el sistema operativo o por apagado o reinicio del dispositivo móvil.

```
[BroadcastReceiver(Enabled = true, Exported = true, DirectBootAware = true, Permission = "android.permission.RECEIVE_BOOT_COMPLETED")]
[IntentFilter(new[] { Intent.ActionBootCompleted,
Intent.ActionReboot, Intent.ActionLockedBootCompleted })]
public class ReceiveMessageService : BroadcastReceiver
{
    public override void OnReceive(Context context, Intent
intent)
    {
        // Start service
        context.StartService(new Intent(context,
typeof(ReceiveService)));
    }
}
```

<sup>28</sup> `BroadcastReceiver`. [Documentación oficial Xamarin Android](#).



Para el logo de la aplicación se han elegido los colores negro y blanco puesto que combinan perfectamente no solo a nivel visual sino también en cuanto a qué sentimiento desprende cada color.



Ilustración 35 - Logo DicereGram

El color negro del logo intenta mostrar fuerza, mientras que el color blanco pretende manifestar pureza y luz.

Se han seleccionado estas sensaciones con respecto a los colores, ya que las personas con discapacidad visual o con cualquier discapacidad suelen ser personas muy fuertes.

Para finalizar con la explicación de los colores y el branding del logo, el atributo luz del color blanco se ha escogido porque siempre debe haber vida en las vidas (valga la redundancia) de todas las personas, pero sobretodo en la vida de las personas con discapacidad visual.

## 7.2. Paleta de colores

La paleta de colores y su respectiva notación HTML seleccionada para la aplicación del proyecto es la siguiente:



Ilustración 36 - Paleta de colores DicereGram

Los colores principales utilizados en la aplicación son el negro (#000000, usado para el color de fondo de toda la aplicación) y el blanco (#ffffff, usado para resaltar los caracteres en la pantalla).

Los siguientes cuatro colores de la paleta se han utilizado para los colores de las diferentes secciones de la aplicación:

- Crimson (#f91d4b): Sección enviar mensaje.
- Celeste (#9cfd7): Sección recibir mensaje/conversaciones.
- Magenta/Fucsia (#fa86da): Sección contactos.
- Verde mar moderado (#87cba4): Sección configuración.

Por último, una mezcla de los últimos dos colores de la paleta se han utilizado únicamente en el menú principal para hacer énfasis sobre el botón de cerrar sesión (Logout).

## 7.3. Mockups

El diseño de los mockups se ha realizado utilizando el software de Balsamiq<sup>30</sup>.


En los mockups mostrados en las secciones siguientes se hace uso de la paleta de colores descrita en la sección anterior. Además, se mostrarán tanto los prototipos de baja fidelidad como los de alta fidelidad desarrollados al principio del proyecto.

Los siguientes mockups se han realizado en paralelo junto con la implementación de las funcionalidades.

### 7.3.1. Mockup login y registro

Como podemos observar en el mockup de Login, se ha hecho uso únicamente del color negro de fondo para resaltar tanto los iconos como los caracteres de los textbox ubicados en la parte central de la pantalla.

Se han utilizado ambos colores ya que las personas con discapacidad visual visualizan mejor los objetos en la pantalla cuando el contraste entre el fondo y dichos objetos es muy alto.

Se ha de destacar el footer incluido en todas las vistas de la aplicación que incluye el nombre de la aplicación junto con el logo, y una marca que incluyo en todas las aplicaciones realizadas por mí (*Made with* ).

En esta vista se le solicitará al usuario (mediante voz) su número de teléfono para poder realizar el login contra Telegram.

Una vez se ha obtenido el número de teléfono y el código de acceso mediante la recepción de un SMS de Telegram, se rellenarán los dos textbox con fondo negro y texto en color blanco ubicados en el centro de la pantalla con dichos datos.

---

<sup>30</sup> Balsamiq. [Sitio web oficial](#).

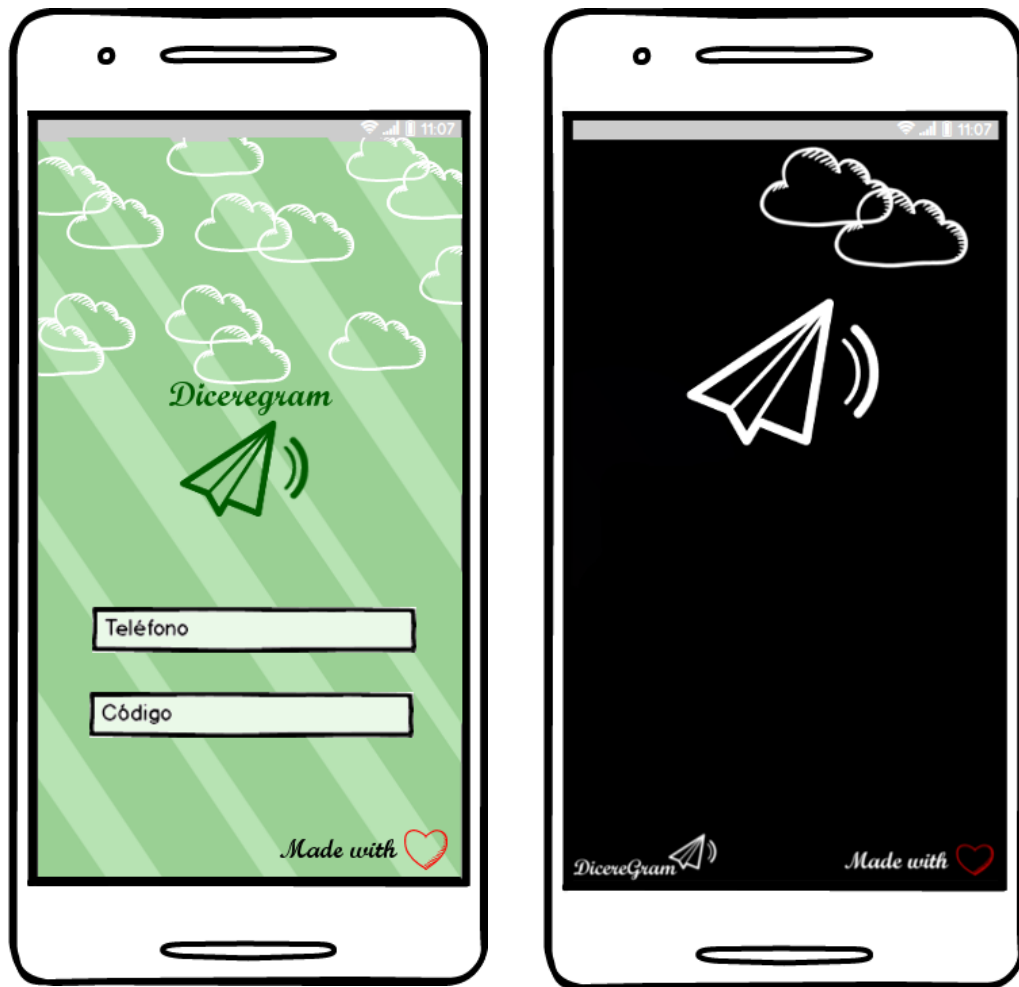


Ilustración 37 - Mockups login y registro

### 7.3.2. Mockup menú principal

Este mockup representa la vista principal de la aplicación. Una vez logueados es la primera vista que nos aparecerá cuando volvamos a abrir la aplicación. Como podemos observar, volvemos a visualizar un contraste alto entre el fondo negro, el cual se repetirá en toda la aplicación, y los diferentes botones que redirigirán a las diferentes secciones de la aplicación.

En esta vista podemos observar el uso de todos los colores mostrados en la paleta de colores.

En esta vista se puede navegar tanto mediante el uso de la voz como interactuando con la pantalla mediante la pulsación de los botones.

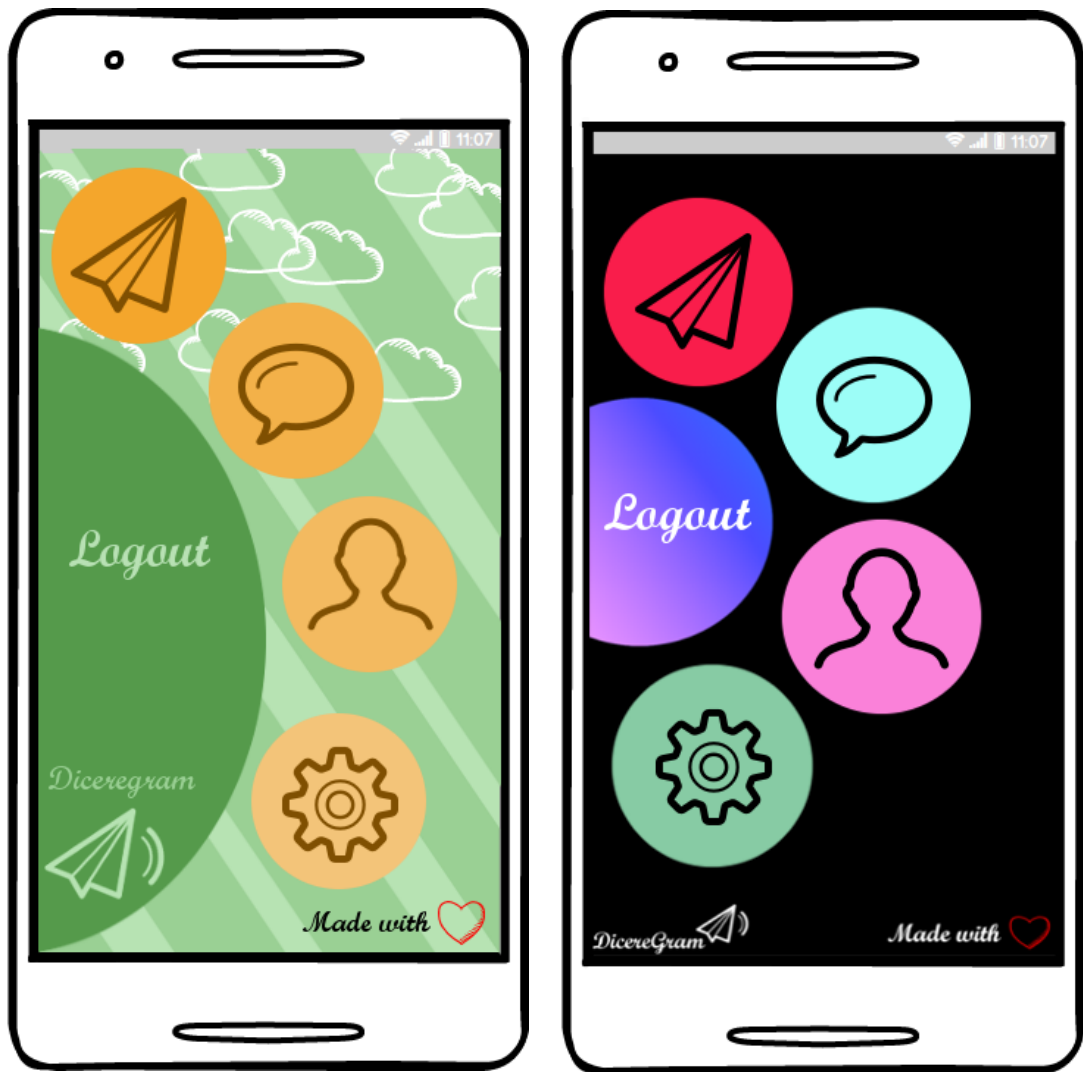


Ilustración 38 - Mockups menú principal

### 7.3.3. Mockup enviar mensaje

En este mockup se representa la vista del envío de mensajes. Se compone de dos textbox con fondo negro y texto en color blanco ubicados a mitad de la pantalla del dispositivo.

En esta vista se solicitará al usuario el contacto a quien enviar el mensaje y el contenido del mensaje que se desea enviar. Una vez obtenidos ambos datos, se rellenarán ambos textbox.

La vista no dispone de ningún botón para realizar/confirmar el envío del mensaje ya que en esta vista la comunicación entre usuario-aplicación se realizará únicamente mediante la voz.

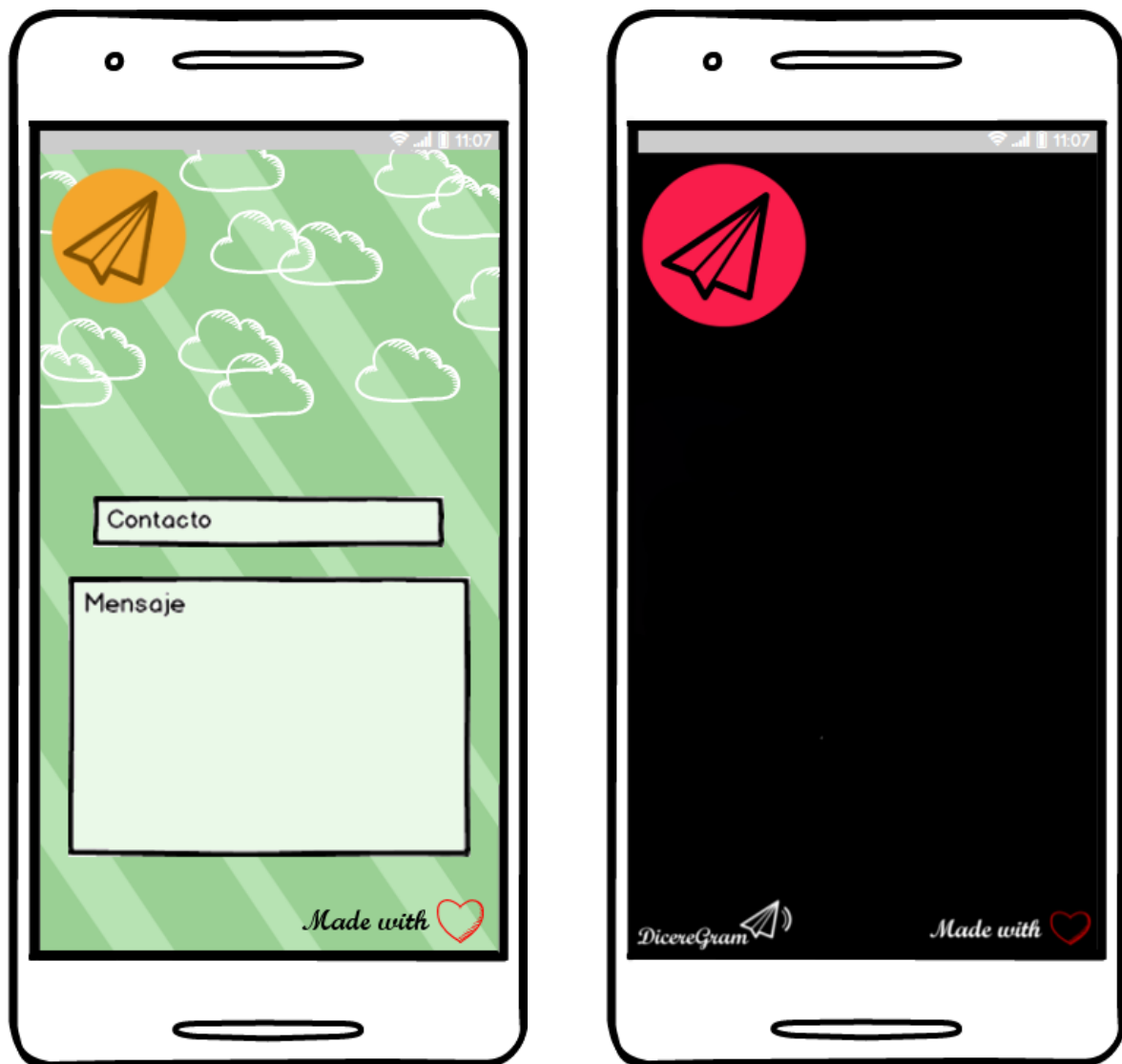


Ilustración 39 - Mockups enviar mensaje

#### 7.3.4. Mockup leer mensajes vista general

Este mockup representa la vista general de los mensajes, es decir, representa una línea por cada conversación con un contacto.

Si se recibe un mensaje mientras el usuario se encuentra en esta vista, la lista de mensajes se actualizará en tiempo real.

Pulsando sobre cualquier de los ítems podremos acceder a la conversación con cada uno de los contactos. Además de poder pulsar sobre los ítems, podremos interactuar con la aplicación a través de la voz para volver atrás, para borrar una conversación o para entrar a la vista de la conversación con el contacto seleccionado.





Ilustración 40 - Mockups mensajes general

### 7.3.5. Mockup leer mensajes vista individual

Este mockup representa la vista individual de una conversación con un contacto, es decir, representa una línea por cada mensaje enviado entre un contacto y el usuario.

Si se recibe un mensaje del contacto del que el usuario se encuentra viendo su conversación, la lista de mensajes se actualizará en tiempo real.

Desde esta vista, el usuario podrá leer los mensajes recibidos sin leer y responder inmediatamente o leer mensajes mediante la búsqueda de una palabra.



Ilustración 41 - Mockup mensajes individual

### 7.3.6. Mockup menú contactos

Este mockup representa la vista del menú de la sección de contactos.

A través de esta vista, el usuario podrá buscar un contacto, actualizar los contactos de Telegram, borrar un contacto y entrar a la sección de bloqueados.

Pulsando sobre cualquier de los ítems podremos acceder a las funcionalidades implementadas para cada uno de ellos, y en el caso de el ítem de bloqueados, podremos acceder a dicha sección. Además de poder pulsar sobre los ítems, podremos interactuar con la aplicación a través de la voz para volver atrás, realizar una acción o para entrar a la vista de contactos bloqueados.



Ilustración 42 - Mockups menú contactos

### 7.3.7. Mockup bloqueo/desbloqueo contactos

Este mockup representa la vista de la lista de los usuarios bloqueados por el usuario.

A través de esta vista, el usuario podrá escuchar los contactos que está bloqueados actualmente y/o bloquear o desbloquear un contacto.

El usuario interactuara con la aplicación en esta vista a través de la voz para volver atrás, realizar una acción o no hacer nada.

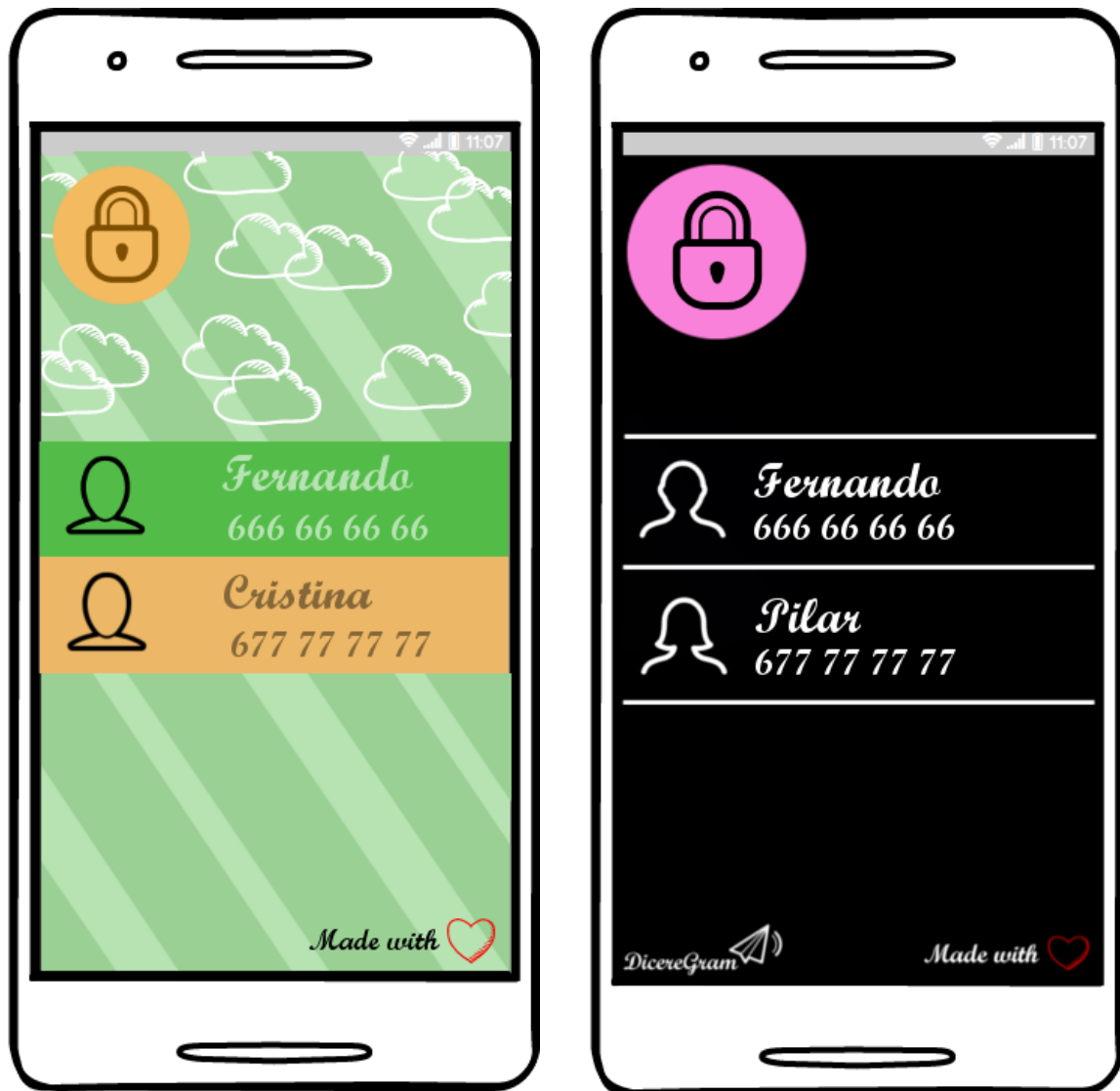


Ilustración 43 - Mockups bloqueo/desbloqueo contactos

### 7.3.8. Mockup Menú Configuración

Este mockup representa la vista del menú de configuración de la aplicación.

A través de esta vista, el usuario podrá modificar su nombre de usuario en Telegram, modificar el tipo de voz, modificar la velocidad de lectura de la voz, y activar o desactivar la voz de la aplicación.

Pulsando sobre cualquier de los ítems podremos acceder a las funcionalidades implementadas para cada uno de ellos. Además de poder pulsar sobre los ítems, podremos interactuar con la aplicación a través de la voz para volver atrás, realizar una acción o para no hacer nada.



Ilustración 44 - Mockups menú configuración

### 7.3.9. Conclusión

El boceto de baja fidelidad no se seleccionó para el prototipo final de la aplicación debido a que la interfaz no cumplía con las características que debe reunir una aplicación enfocada a personas con discapacidad visual, por ejemplo:

- No hay un contraste alto entre el fondo de primer plano y el texto.
- No se utilizan colores oscuros de fondo y colores claros para el texto.
- Los colores seleccionados no ayudan a una mejor visualización de los componentes de la pantalla.

## 7.4. Diagrama de navegación

Se ha realizado un diagrama de navegación donde se muestra gráficamente la navegación que puede realizar el usuario a través de las diferentes vistas que componen la aplicación.

DIAGRAMA NAVEGACIÓN DICEREGRAM

Paula Frias Arroyo | December 14, 2017

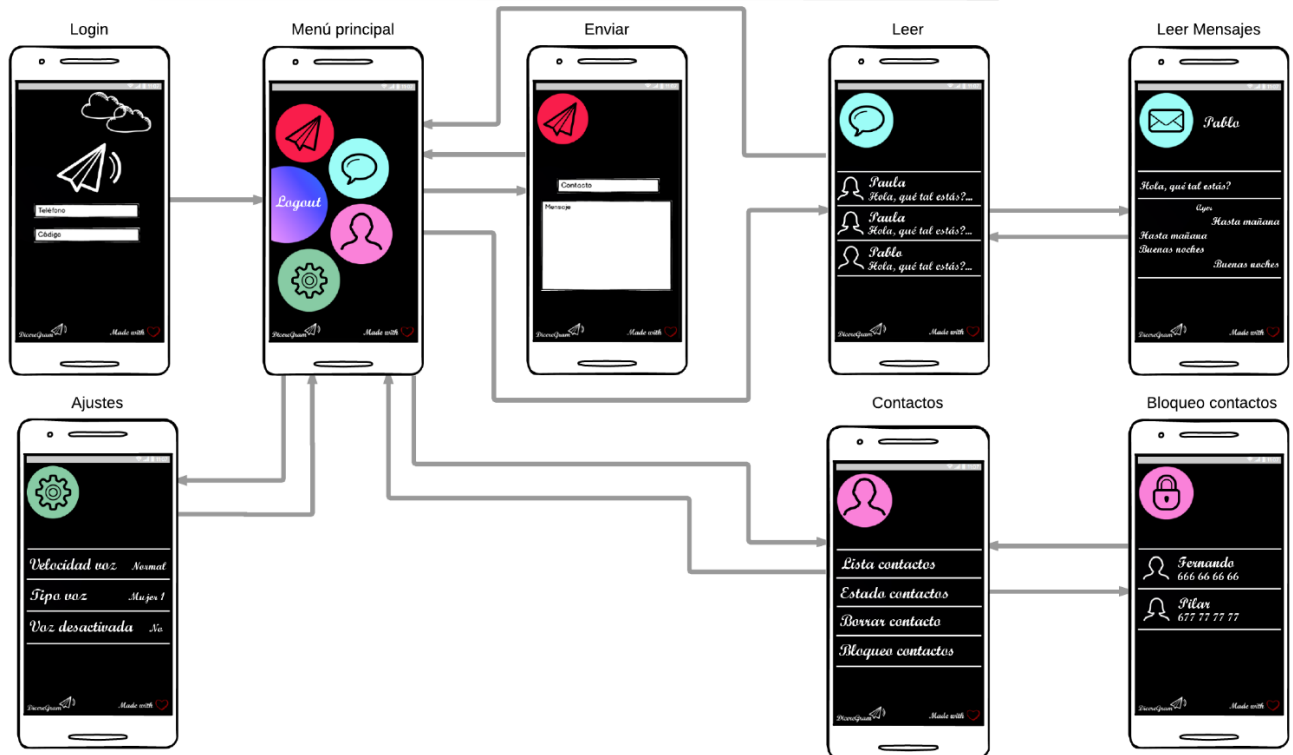


Ilustración 45 - Diagrama de navegación DicereGram

# 8. Funcionalidades

## 8.1. Diagrama de casos de uso

Se ha realizado un diagrama de casos de uso comprendiendo todos los casos de uso que abarcará la aplicación DicereGram.

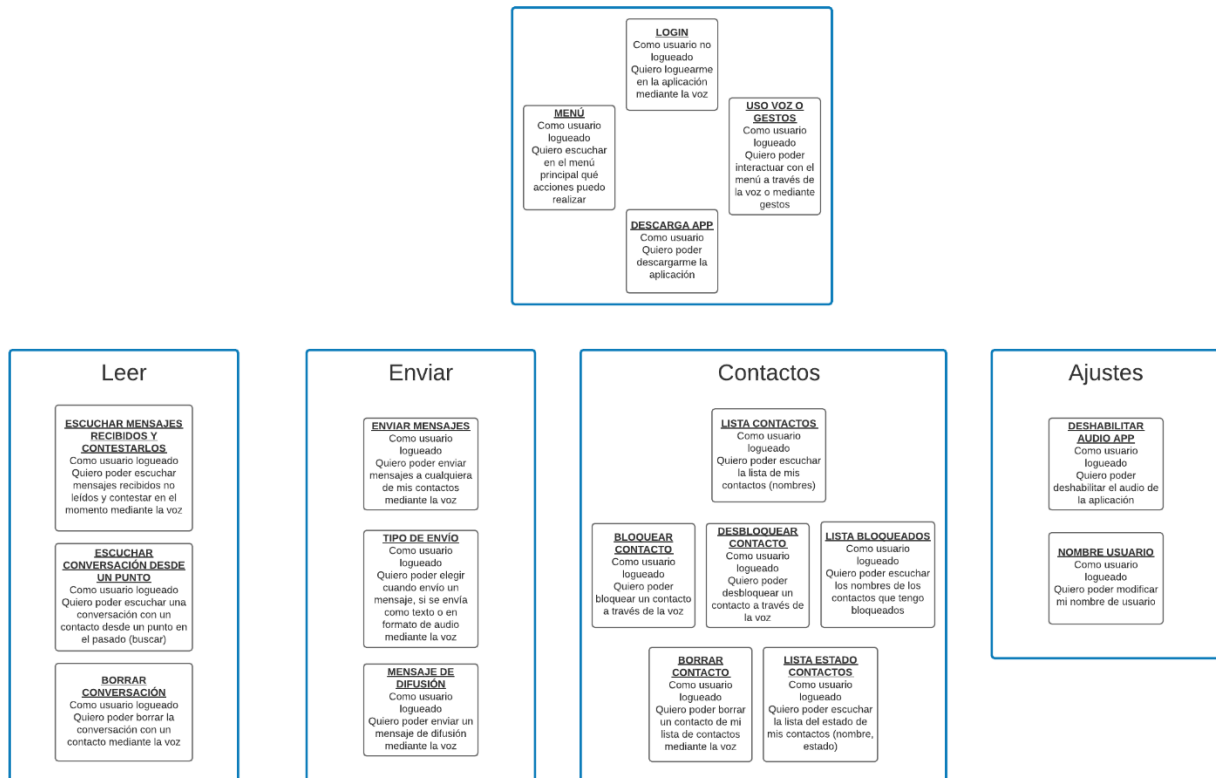


Ilustración 46 - Diagrama de casos de uso DicereGram

## 8.2. Funcionalidades

Las funcionalidades implementadas en la aplicación DicereGram se realizan mediante la comunicación entre usuario-aplicación haciendo uso de la voz. Las funcionalidades son las siguientes:

- Login y registro:
  - ❖ La seguridad empleada se realiza a través del protocolo de Telegram, MTProto, en el que se profundizará en secciones posteriores.

- ❖ La primera vez que se lance la aplicación DicereGram, se solicitará al usuario su número de teléfono y se realizará el registro en Telegram si el usuario no tiene cuenta creada.
- ❖ Tras esto, se procederá a la solicitud del código de acceso a Telegram. Éste es un código de 5 dígitos que el usuario recibirá a través de un SMS.
- ❖ Si se produce cualquier error durante el login, se comunicará al usuario que ha ocurrido un error y que lo intente más tarde. En caso contrario, se creará la sesión del usuario en el dispositivo y las credenciales del usuario para el login de Telegram quedarán almacenadas en la base de datos local a su dispositivo móvil creada para ello.
- Envío de mensajes:
  - ❖ Envío de mensajes de texto:
    - El usuario podrá enviar un mensaje de texto a uno de sus contactos.
    -
- Lectura de mensajes:
  - ❖ Recepción de mensajes recibidos en la aplicación Telegram:
    - Se ha realizado la creación de un servicio que estará a la escucha de los mensajes entrantes del usuario.
    - Cuando se recibe un mensaje, se avisa al usuario a través de una notificación, y si se encuentra con la aplicación DicereGram en funcionamiento, se recargarán las vistas de las conversaciones automáticamente.
  - ❖ Lectura de mensajes recibidos:
    - Los mensajes recibidos será leídos haciendo uso de la voz una vez el usuario acceda a la vista de conversaciones, y, además, tenga mensajes nuevos sin leer.
    -
  - ❖ Búsqueda de texto en las conversaciones:



- Se le permitirá al usuario, una vez acceda a una conversación con un contacto, buscar por una palabra o texto en dicha conversación.
  -
- Contactos:
  - ❖ Buscar un contacto:
    - El usuario podrá buscar un contacto almacenado en Telegram. La aplicación usará la voz para leerle al usuario los datos del contacto a buscar: nombre, teléfono, estado y si está bloqueado o no.
  - ❖ Actualizar contactos:
    - El usuario podrá actualizar los contactos que tiene almacenados tanto en Telegram como en DicereGram, haciendo uso de la voz para comunicarse con la aplicación DicereGram.
  - ❖ Borrar un contacto:
    - El usuario podrá borrar cualquiera de los contactos de la aplicación.
  - ❖ Contactos bloqueados:
    - Visualizar mis contactos bloqueados:
      - El usuario podrá observar la lista de los contactos que tiene bloqueados, y, además, la aplicación leerá sus nombres haciendo uso de la síntesis de voz.
    - Bloquear/desbloquear un contacto:
      - El usuario podrá bloquear un contacto de los que tiene almacenados en DicereGram, o desbloquear un contacto de los que tiene bloqueados.
- Configuración:
  - ❖ Modificar tipo de voz:
    - El usuario podrá modificar el tipo de voz de la síntesis de voz, si tiene más de un tipo de voz instalado.

- ❖ Modificar velocidad de lectura de la voz:
  - El usuario podrá tanto aumentar como reducir la velocidad de la síntesis de voz usada en la aplicación.
- ❖ Modificar nombre de usuario en Telegram:
  - El usuario podrá modificar su nombre de usuario para la aplicación de Telegram.
- ❖ Activar/desactivar voz:
  - El usuario podrá activar o desactivar el uso de la síntesis de voz y del reconocimiento de voz.

## 8.3. Interacción conversaciones

Hemos decidido ser más exhaustivos en cuanto a mostrar todas las posibles conversaciones que el usuario puede mantener con la aplicación DicereGram, puesto que la funcionalidad principal de esta es interactuar con el usuario a través de la voz.

Estas conversaciones se han realizado antes del desarrollo del proyecto y se han utilizado durante este simulando un proceso iterativo.

Las posibilidades con respecto a las conversaciones mantenidas con el usuario en cada una de las secciones de la aplicación se describen en los siguientes apartados.

Únicamente se mostrará la primera parte de cada una de las conversaciones implementadas en la aplicación DicereGram. Si se desea visualizar todas las posibles conversaciones que la aplicación puede mantener con el usuario, en el anexo de este documento se encontrarán las posibles conversaciones restantes.

## 8.3.1. Registro y Login

### DICEREGRAM

#### CONVERSACIÓN LOGIN

##### Primera posibilidad

**DicereGram** ¡Bienvenido a DicereGram! ¿Cuál es su número de teléfono?  
**Usuario** 666666666  
**DicereGram** Login Correcto  
**DicereGram** Redirección a Menú principal (no hay voz)

##### Segunda posibilidad

**DicereGram** ¡Bienvenido a DicereGram! ¿Cuál es su número de teléfono?  
**Usuario** 666  
**DicereGram** El número de teléfono no tiene un formato correcto. Por favor, diga un número de teléfono válido  
  
**Usuario** 666666666  
**DicereGram** Login Correcto  
**DicereGram** Redirección a Menú principal (no hay voz)

##### Tercera posibilidad

**DicereGram** ¡Bienvenido a DicereGram! ¿Cuál es su número de teléfono?  
**Usuario** 666666666  
**DicereGram** No dispone de una cuenta en Telegram. ¿Cuál quiere que sea su nombre de usuario?  
**Usuario** Paula  
**DicereGram** Login Correcto  
**DicereGram** Redirección a Menú principal (no hay voz)

##### Cuarta posibilidad

**DicereGram** ¡Bienvenido a DicereGram! ¿Cuál es su número de teléfono?  
**Usuario** 666666666  
**DicereGram** Lo siento no le he entendido. ¿Cuál es su número de teléfono?  
**Usuario** 666666666  
**DicereGram** Login Correcto  
**DicereGram** Redirección a Menú principal (no hay voz)

##### Quinta posibilidad

**DicereGram** ¡Bienvenido a DicereGram! ¿Cuál es su número de teléfono?  
**Usuario** 666666666  
**DicereGram** Ha ocurrido un error en la recepción del SMS. Por favor, inténtelo en aproximadamente media hora.

##### Sexta posibilidad

**DicereGram** ¡Bienvenido a DicereGram! ¿Cuál es su número de teléfono?  
**Usuario** 666666666  
**DicereGram** Ha ocurrido un error al realizar el login contra Telegram. Por favor, inténtelo en aproximadamente dos horas.

*Ilustración 47 - Conversaciones registro y login*

## 8.3.2. Menú principal

### DICEREGRAM

#### CONVERSACIÓN MENÚ PRINCIPAL

##### Primera posibilidad

**DicereGram** ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?

**Usuario** Enviar

**DicereGram** Redirección a Enviar (no hay voz)

##### Segunda posibilidad

**DicereGram** ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?

**Usuario** x

**DicereGram** Lo siento, no le he entendido. ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión o no hacer nada?

**Usuario** Enviar

**DicereGram** Redirección a Enviar (no hay voz)

##### Tercera posibilidad

**DicereGram** ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?

**Usuario** Nada

##### Cuarta posibilidad

**DicereGram** ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?

**Usuario** x

**DicereGram** Lo siento, no ha dicho una opción válida. ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión o no hacer nada?

**Usuario** Contactos

**DicereGram** Redirección a Menú contactos (no hay voz)

##### Quinta posibilidad

**DicereGram** ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?

**Usuario** Cerrar sesión

**DicereGram** ¿Está seguro que quiere cerrar la sesión? Hasta dentro de 2 horas no podrá volverse a logear con el mismo número de teléfono.

**Usuario** Sí

**DicereGram** Sesión cerrada correctamente.

*Ilustración 48 – Conversaciones menú principal*

### 8.3.3. Enviar mensaje

#### DICEREGRAM

##### CONVERSACIÓN ENVIAR

###### Primera posibilidad

**DicereGram** Diga el nombre del contacto a quien quiere enviar un mensaje  
**Usuario** Pablo  
**DicereGram** ¿Qué mensaje quiere enviar?  
**Usuario** Hola, qué tal estás  
**DicereGram** Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?  
**Usuario** Sí  
**DicereGram** Mensaje enviado  
**DicereGram** Redirección a MainActivity (no hay voz)

###### Segunda posibilidad

**DicereGram** Diga el nombre del contacto a quien quiere enviar un mensaje  
**Usuario** Pablo  
**DicereGram** Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto a quien quiere enviar un mensaje  
**Usuario** Pablo  
**DicereGram** ¿Qué mensaje quiere enviar?  
**Usuario** Hola, qué tal estás  
**DicereGram** Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?  
**Usuario** Sí  
**DicereGram** Mensaje enviado  
**DicereGram** Redirección a MainActivity (no hay voz)

###### Tercera posibilidad

**DicereGram** Diga el nombre del contacto a quien quiere enviar un mensaje  
**Usuario** Pepe  
**DicereGram** Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea enviar un mensaje, diga terminar  
**Usuario** Pablo  
**DicereGram** ¿Qué mensaje quiere enviar?  
**Usuario** Hola, qué tal estás  
**DicereGram** Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?  
**Usuario** Sí  
**DicereGram** Mensaje enviado  
**DicereGram** Redirección a MainActivity (no hay voz)

###### Cuarta posibilidad

**DicereGram** Diga el nombre del contacto a quien quiere enviar un mensaje  
**Usuario** Pablo  
**DicereGram** ¿Qué mensaje quiere enviar?  
**Usuario** Hola, qué tal estás  
**DicereGram** Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?  
**Usuario** Sí  
**DicereGram** No se ha podido enviar el mensaje. Inténtelo más tarde  
**DicereGram** Redirección a MainActivity (no hay voz)

*Ilustración 49 - Conversaciones enviar mensaje*

### 8.3.4. Menú contactos

#### DICEREGRAM

##### CONVERSACIÓN MENÚ CONTACTOS

###### Primera posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Buscar

**DicereGram** Redirección a ContactosNombre (no hay voz)

###### Segunda posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, o no hacer nada?

**Usuario** x

**DicereGram** Lo siento, no le he entendido. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, o no hacer nada?

**Usuario** Buscar

**DicereGram** Redirección a ContactosNombre (no hay voz)

###### Tercera posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Actualizar

**DicereGram** Contactos actualizados. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Buscar

**DicereGram** Redirección a ContactosNombre (no hay voz)

###### Cuarta posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Eliminar

**DicereGram** Diga el nombre del contacto que quiere eliminar

**Usuario** Pablo

**DicereGram** Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto a quien quiere enviar un mensaje

**Usuario** Pablo

**DicereGram** ¿Está seguro que quiere eliminar a Pablo?

**Usuario** Sí

**DicereGram** Contacto eliminado. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Buscar

**DicereGram** Redirección a ContactosNombre (no hay voz)

*Ilustración 50 - Conversaciones menú contactos*

## 8.3.5. Buscar contacto

### DICEREGRAM

#### CONVERSACIÓN BUSCAR CONTACTOS

##### Primera posibilidad

**DicereGram** Diga el nombre del contacto que quiere buscar  
**Usuario** Pablo  
**DicereGram** El contacto buscado es: Nombre: Pablo Corral. Teléfono: x. Estado: x. Bloqueado: No está bloqueado.  
**DicereGram** ¿Quiere buscar otro contacto, no hacer nada o volver atrás?  
**Usuario** Nada

##### Segunda posibilidad

**DicereGram** Diga el nombre del contacto que quiere buscar  
**Usuario** Pablo  
**DicereGram** Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto que desea buscar  
**Usuario** Pablo  
**DicereGram** El contacto buscado es: Nombre: Pablo Corral. Teléfono: x. Estado: x. Bloqueado: No está bloqueado.  
**DicereGram** ¿Quiere buscar otro contacto, no hacer nada o volver atrás?  
**Usuario** Nada

##### Tercera posibilidad

**DicereGram** Diga el nombre del contacto que quiere buscar  
**Usuario** Pepe  
**DicereGram** Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea buscar un contacto, diga terminar  
**Usuario** Pablo  
**DicereGram** El contacto buscado es: Nombre: Pablo Corral. Teléfono: x. Estado: x. Bloqueado: No está bloqueado.  
**DicereGram** ¿Quiere buscar otro contacto, no hacer nada o volver atrás?  
**Usuario** Volver atrás  
**DicereGram** Redirección a Menú contactos (no hay voz)

##### Cuarta posibilidad

**DicereGram** Diga el nombre del contacto que quiere buscar  
**Usuario** Pepe  
**DicereGram** Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea buscar un contacto, diga terminar  
**Usuario** Terminar  
**DicereGram** Redirección a Menú contactos (no hay voz)

##### Quinta posibilidad

**DicereGram** Diga el nombre del contacto que quiere buscar  
**Usuario** x  
**DicereGram** Lo siento, no le he entendido. ¿Quiere buscar un contacto, no hacer nada o volver atrás?  
**Usuario** Nada

*Ilustración 51 - Conversaciones buscar contacto*

## 8.3.6. Bloquear/Desbloquear un contacto

### DICEREGRAM

#### CONVERSACIÓN BLOQUEAR/DESBLOQUEAR CONTACTOS

##### Primera posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Bloquear
<b>DicereGram</b>	Diga el nombre del contacto que quiere bloquear
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Contacto bloqueado
<b>DicereGram</b>	¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Segunda posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Desbloquear
<b>DicereGram</b>	Diga el nombre del contacto que quiere desbloquear
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto que desea bloquear/desbloquear
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Contacto desbloqueado
<b>DicereGram</b>	¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Tercera posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Bloquear
<b>DicereGram</b>	Diga el nombre del contacto que quiere bloquear
<b>Usuario</b>	Pepe
<b>DicereGram</b>	Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea bloquear o desbloquear un contacto, diga terminar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Contacto bloqueado
<b>DicereGram</b>	¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?

##### Cuarta posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Bloquear
<b>DicereGram</b>	Diga el nombre del contacto que quiere bloquear
<b>Usuario</b>	Pepe
<b>DicereGram</b>	Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea bloquear o desbloquear un contacto, diga terminar
<b>Usuario</b>	Terminar
<b>DicereGram</b>	Redirección a MisContactos (no hay voz)

*Ilustración 52 - Conversaciones bloquear/desbloquear contacto*



## 8.3.7. Mensajes general

### DICEREGRAM

#### CONVERSACIÓN LEER MENSAJES GENERAL

##### Primera posibilidad

- DicereGram** Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
- Usuario** Entrar a una conversación
- DicereGram** Por favor, diga el nombre del contacto
- Usuario** Pablo
- DicereGram** Redirección a SingleChat de Pablo (no hay voz)

##### Segunda posibilidad

- DicereGram** Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
- Usuario** Entrar a una conversación
- DicereGram** Por favor, diga el nombre del contacto
- Usuario** Pablo
- DicereGram** Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto del que desea entrar a la conversación
- Usuario** Pablo
- DicereGram** Redirección a SingleChat de Pablo (no hay voz)

##### Tercera posibilidad

- DicereGram** Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
- Usuario** x
- DicereGram** Lo siento, no le he entendido. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?

##### Cuarta posibilidad

- DicereGram** Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
- Usuario** Borrar
- DicereGram** Por favor, diga el nombre del contacto
- Usuario** Pablo
- DicereGram** Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto del que desea entrar a la conversación
- Usuario** Pablo
- DicereGram** Conversación con Pablo borrada. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?

*Ilustración 53 - Conversaciones mensajes general*

## 8.3.8. Mensajes chat individual

### DICEREGRAM

#### CONVERSACIÓN LEER MENSAJES INDIVIDUAL

##### Primera posibilidad

<b>DicereGram</b>	Los mensajes nuevos de Pablo son: ... ¿Quiere responder?
<b>Usuario</b>	Sí
<b>DicereGram</b>	¿Qué mensaje quiere enviar?
<b>Usuario</b>	Holi
<b>DicereGram</b>	Su mensaje es Holi. ¿Quiere enviarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	Mensaje enviado. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Segunda posibilidad

<b>DicereGram</b>	Los mensajes nuevos de Pablo son: ... ¿Quiere responder?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Tercera posibilidad

<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Mensaje
<b>DicereGram</b>	Diga el mensaje desde la que quiere leer
<b>Usuario</b>	Xamarin
<b>DicereGram</b>	He terminado. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Cuarta posibilidad

<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Mensaje
<b>DicereGram</b>	Diga el mensaje desde la que quiere leer
<b>Usuario</b>	Xamarin
<b>DicereGram</b>	No existen mensajes con esa palabra. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a MisChats (no hay voz)

##### Quinta posibilidad

<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Buscar por mensaje
<b>DicereGram</b>	¿Qué mensaje quiere buscar?
<b>Usuario</b>	Xamarin
<b>DicereGram</b>	Tú: ... Pablo: ... He terminado. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

Ilustración 54 - Conversaciones chat individual

## 8.3.9. Menú configuración

### DICEREGRAM

#### CONVERSACIÓN MENÚ CONFIGURACIÓN

##### Primera posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Modificar nombre de usuario
<b>DicereGram</b>	Su nombre de usuario es X. ¿Quiere modificarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	¿Qué nombre de usuario quiere?
<b>Usuario</b>	paufri
<b>DicereGram</b>	Su nombre de usuario ha sido modificado correctamente a paufri. ¿Quiere modificar tu nombre de usuario, el tipo de voz, la velocidad de lectura o la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Segunda posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Modificar nombre de usuario
<b>DicereGram</b>	Su nombre de usuario es X. ¿Quiere modificarlo?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

##### Tercera posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Modificar nombre de usuario
<b>DicereGram</b>	Su nombre de usuario es X. ¿Quiere modificarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	¿Qué nombre de usuario quiere?
<b>Usuario</b>	paufri
<b>DicereGram</b>	El nombre de usuario no es válido. Por favor, diga otro nombre de usuario o cancelar
<b>Usuario</b>	Cancelar
<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

Ilustración 55 - Conversaciones menú configuración

# 9. Seguridad

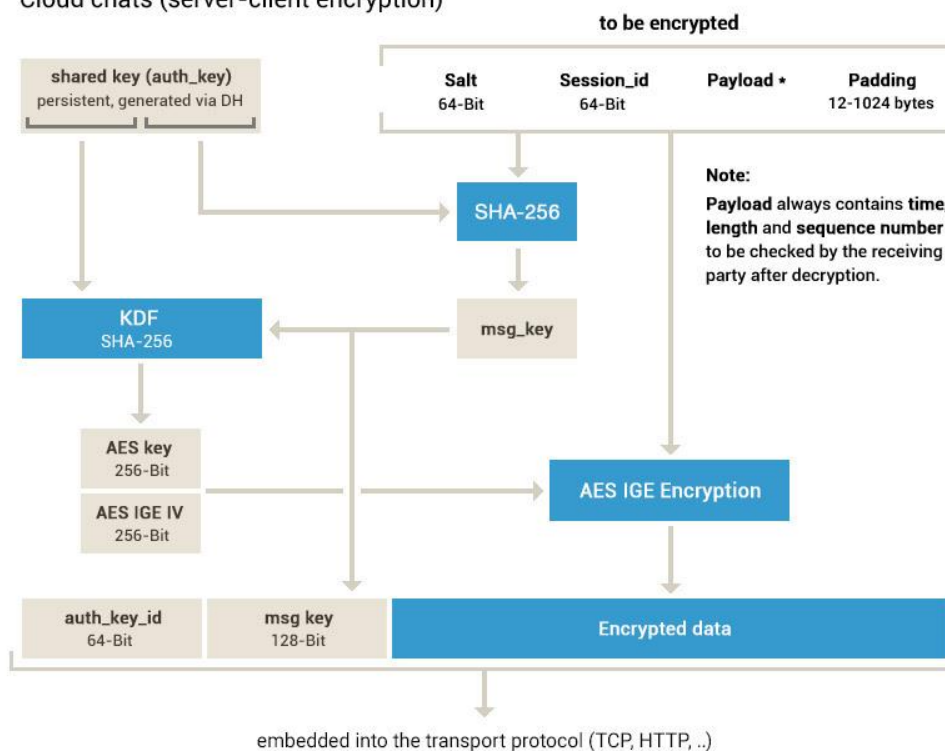
## 9.1. MTPROTO

Mobile Transport Protocol<sup>31</sup> (MTPROTO) es el nombre del protocolo de datos de Telegram Messenger. Está desarrollado bajo un estándar abierto, a base de API Java, desarrollado por el matemático Nikolái Dúrov y financiado por Pável Dúrov. Fue anunciado en 2013 por la antes empresa estadounidense Digital Fortress, ahora desarrollada por Telegram Messenger LLP.

A diferencia de XMPP, MTPROTO está enfocado en la multisesión multiplataforma y el transporte de archivos sin importar su formato o capacidad. El tráfico tiene dos tipos de cifrados, ambos con AES de base, siendo los chat secretos para el envío de mensajes de forma restrictiva contra la filtración de información sensible. Es una de las pocas aplicaciones desarrolladas fuera de los Estados Unidos tras el incidente del ciberespionaje.

### MTPROTO 2.0, part I

Cloud chats (server-client encryption)



**Important:** After decryption, the receiver must check that  $\text{msg\_key} = \text{SHA-256}(\text{fragment of auth\_key} + \text{decrypted data})$

Ilustración 56 - Estructura MTPROTO

<sup>31</sup> MTPROTO. [Sitio web oficial.](#)

El protocolo está diseñado para acceder a una API de servidor desde aplicaciones que se ejecutan en dispositivos móviles. Debe enfatizarse que un navegador web no es una aplicación de este tipo.

El protocolo se subdivide en tres componentes virtualmente independientes:

- Componente de alto nivel (lenguaje de consulta de API): define el método mediante el cual las consultas y respuestas de API se convierten en mensajes binarios.
- Capa criptográfica (autorización): define el método por el cual los mensajes se cifran antes de transmitirse a través del protocolo de transporte.
- Componente de transporte: define el método para que el cliente y el servidor transmitan mensajes a través de algún otro protocolo de red existente (como HTTP, HTTPS, TCP, UDP).

## 9.2. GDPR (General Data Protection Regulation)

El Reglamento General de Protección de Datos<sup>32</sup> (RGPD) es la nueva normativa que regula la protección de los datos de los ciudadanos que vivan en la Unión Europea.

El reglamento entró en vigor el 24 de mayo de 2016, pero será de obligado cumplimiento a partir del 25 de mayo de 2018.

Se trata de la primera norma sobre esta materia que afecta a todos los países de la Unión Europea y unifica, por tanto, tanto los derechos como las obligaciones.

Esta nueva normativa determina que todas las empresas, independientemente de su país de origen o de actividad, deberán cumplirla si recogen, guardan, tratan, usan o gestionan algún tipo de dato de los ciudadanos de la Unión Europea. Y, por supuesto, nos afecta a todas las personas que vivimos en la Unión Europea.

La GDPR recoge y reconoce, por tanto, derechos, como al olvido y el derecho a la portabilidad.

El derecho al olvido establece que los ciudadanos podemos solicitar y lograr que nuestros datos personales sean eliminados.

---

<sup>32</sup> GDPR. [Sitio web oficial](#).

Mientras, el derecho a la portabilidad te permite que, si tus datos se están tratando de manera automatizada, puedas recuperarlos en un formato para cederlos a otro responsable.

Un dato personal es cualquier información relacionada con una persona física que se puede utilizar para identificarla directa o indirectamente.

Puede ser cualquier cosa: desde un nombre, una foto, una dirección de correo electrónico, datos bancarios, publicaciones en sitios web de redes sociales, información médica o una dirección IP de un ordenador.

El proyecto DicereGram no ha contemplado ningún tipo de protección de datos, ya que el protocolo propio de Telegram (MTProto) es el encargado de proteger cada dato enviado a través de la API de Telegram. Además, no se almacena ningún dato del usuario fuera de su propio teléfono móvil, es decir, todos los datos relevantes del usuario tales como número de teléfono o código único de Telegram sólo son almacenados en el teléfono móvil del usuario.

# 10. Testing

---

## 10.1. NUnit

NUnit<sup>33</sup> es un marco de pruebas unitarias de código abierto para todos los lenguajes .NET. Inicialmente portado desde JUnit<sup>34</sup>, la versión de producción actual, versión 3, ha sido completamente reescrita con muchas características nuevas y soporte para una amplia gama de plataformas .NET.



*Ilustración 57 - Logo NUnit*

Para poder implementar NUnit en el proyecto DicereGram se ha hecho uso del paquete Nuget.

### 10.1.1. Tipo de pruebas unitarias realizadas

Las pruebas unitarias realizadas durante el desarrollo del proyecto DicereGram han sido implementación y test.

En cuanto se desarrollaba una nueva funcionalidad (método de repositorio), se implementaban los test que testeaban dicha funcionalidad y se volvían a pasar todos los test.

### 10.1.2. Vista y ejecución en Xamarin

Para poder realizar la ejecución de los test unitarios implementados con NUnit, en primer lugar es necesario establecer el proyecto de pruebas como proyecto de inicio.

---

<sup>33</sup> NUnit. [Sitio web oficial](#).

<sup>34</sup> JUnit. [Sitio web oficial](#).

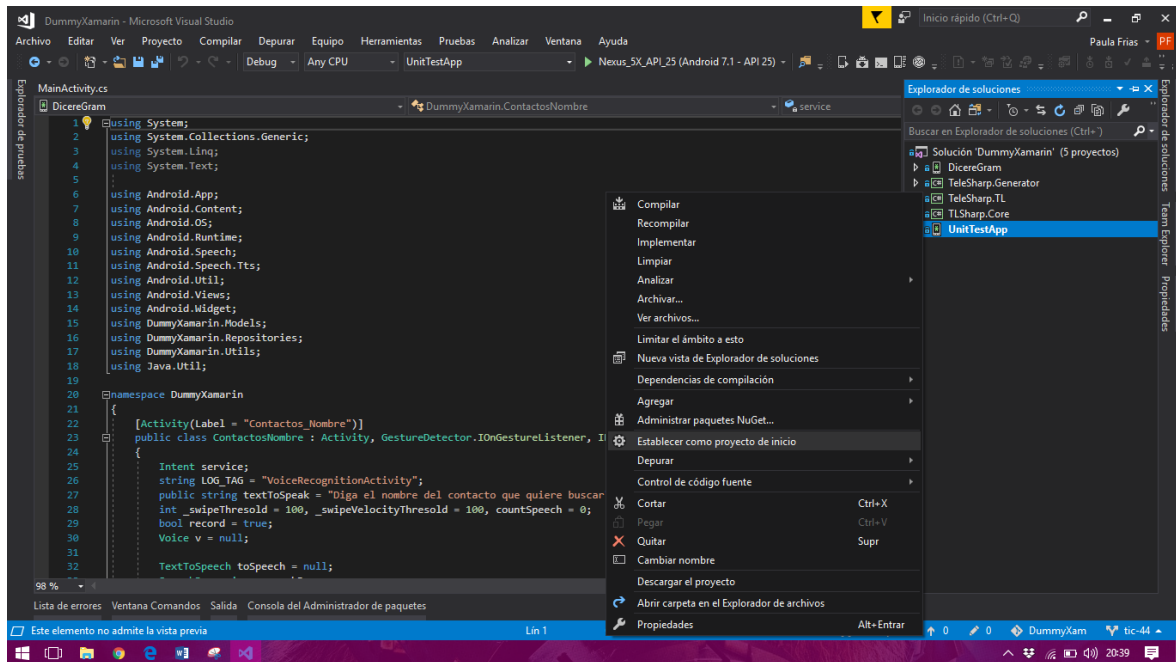


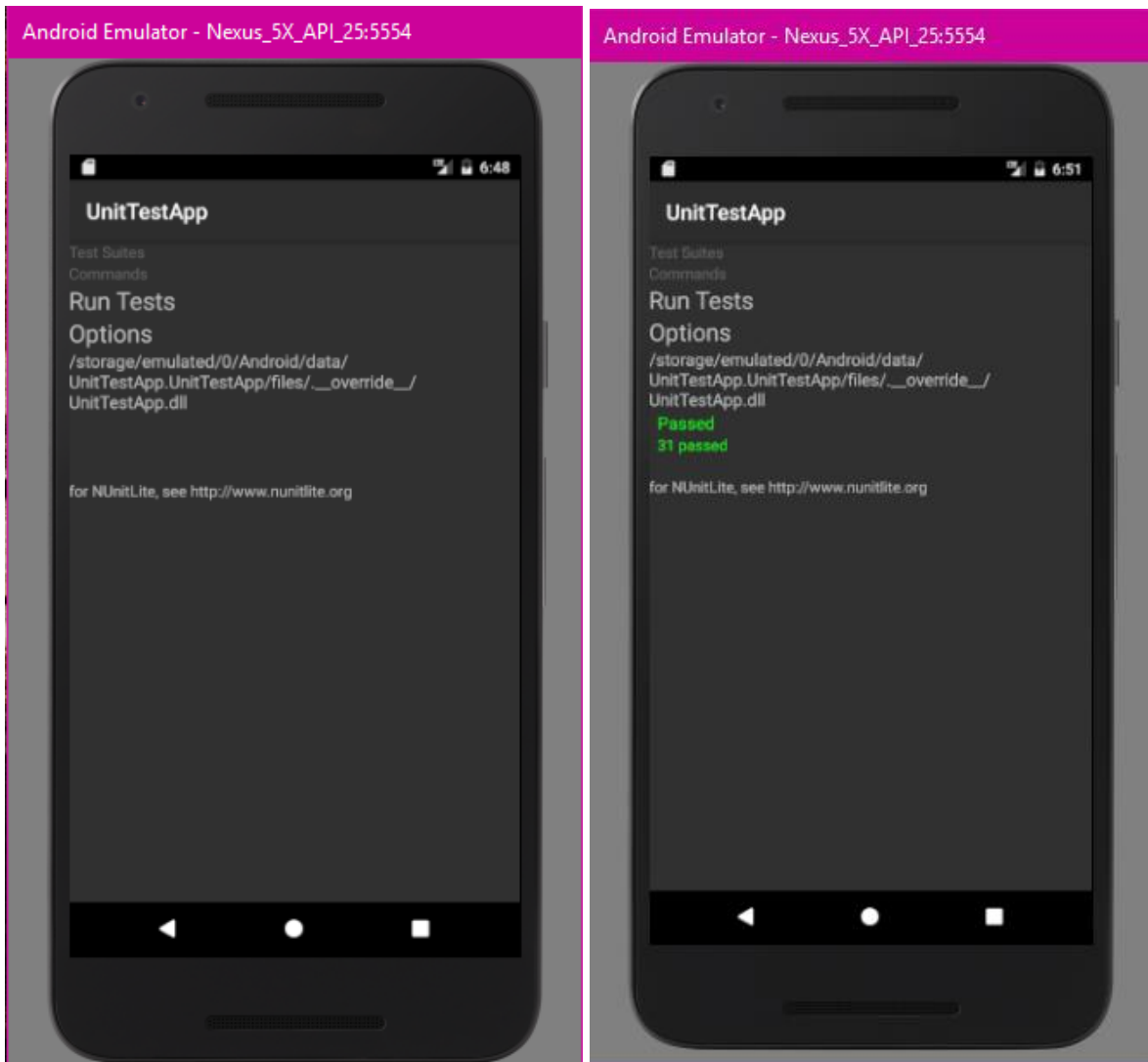
Ilustración 58 - Visual Studio 2017

A continuación, desde el emulador de Android, previamente instalado, ejecutaremos el proyecto de pruebas establecido como proyecto de inicio en el paso anterior.

Una vez completamente ejecutado, se nos mostrará el emulador de Android con la interfaz de usuario de NUnit.

Para comenzar con la ejecución de los test, pulsamos sobre el botón “Run Tests”, y la vista resultante, una vez finalizados todos los tests unitarios, se mostrará el resultado de los tests:





*Ilustración 59 - Vistas ejecución test unitarios NUnit*

### 10.1.3. Test unitarios

Se han implementado tests unitarios para cada uno de los métodos desarrollados en los repositorios del proyecto.

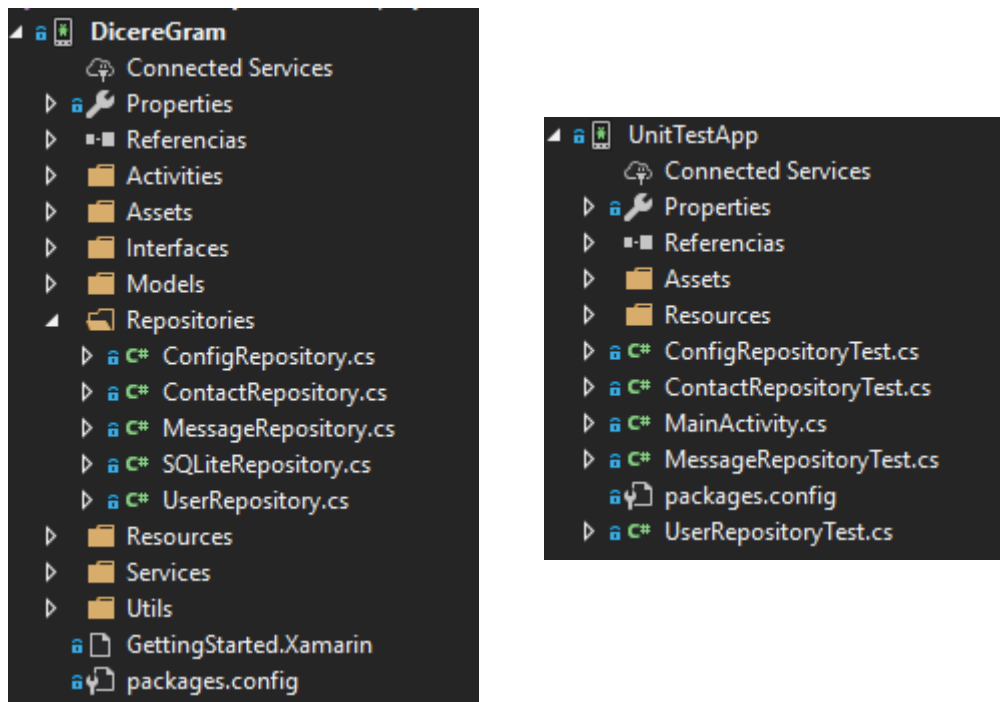


Ilustración 60 - Estructura física proyectos DicereGram

En todas las clases de los test unitarios se ha implementado el método TearDown para restaurar la base de datos a un estado controlado para la correcta ejecución de las pruebas unitarias.

```
[TestFixture]
public class MessageRepositoryTest
{
    [SetUp]
    public void Setup()
    {
    }

    [TearDown]
    public void Tear()
    {
        SQLiteRepository con = new SQLiteRepository();
        con.DeleteDataDatabase();
    }
}
```

Las verificaciones en las pruebas unitarias implementadas se han realizado haciendo uso de Assertions<sup>35</sup>.

<sup>35</sup> Asserts. [Descripción](#).

A continuación se muestra uno de los trozos de código de ejemplo más relevante de los tests unitarios implementados:

```
[Test]
public void GetMessagesOrdered()
{
    SQLiteRepository con = new SQLiteRepository();
    con.CreateDatabase();
    MessageRepository messageRepository = new
MessageRepository(con);
    ContactRepository contactRepository = new
ContactRepository(con);
    Contact contact = new Contact
    {
        Id = 1,
        FirstName = "Pablo",
        LastName = "Corral",
        Phone = "34676681420"
    };
    Contact c = new Contact
    {
        Id = 2,
        FirstName = "Cristina",
        LastName = "Gambin",
        Phone = "34666666666"
    };
    Contact c1 = new Contact
    {
        Id = 3,
        FirstName = "Zoe",
        LastName = "",
        Phone = "34677777777"
    };
    Chat m = new Chat
    {
        Mensaje = "hola",
        FromTo = "34676681420",
        Send = true,
        Created = DateTime.Now.AddDays(-4).AddHours(-5),
        Seen = false
    };
    Chat m1 = new Chat
    {
        Mensaje = "hola, que tal?",
        FromTo = "34676681420",
        Send = false,
        Created = DateTime.Now.AddDays(-3).AddHours(-4),
        Seen = false
    };
}
```

```

};
Chat m2 = new Chat
{
    Mensaje = "soy pablo",
    FromTo = "34676681420",
    Send = false,
    Created = DateTime.Now.AddDays(-3).AddHours(-2),
    Seen = false
};
Chat m3 = new Chat
{
    Mensaje = "hola, que tal pau?",
    FromTo = "34666666666",
    Send = false,
    Created = DateTime.Now.AddDays(-3),
    Seen = false
};
Chat m4 = new Chat
{
    Mensaje = "soy cristina",
    FromTo = "34666666666",
    Send = false,
    Created = DateTime.Now.AddDays(-2),
    Seen = false
};
Chat m5 = new Chat
{
    Mensaje = "soy zoe",
    FromTo = "34677777777",
    Send = false,
    Created = DateTime.Now.AddDays(-2),
    Seen = false
};
Chat m6 = new Chat
{
    Mensaje = "holii",
    FromTo = "34677777777",
    Send = false,
    Created = DateTime.Now.AddDays(-1),
    Seen = false
};
using (var connection = con.GetConnection())
{
    contactRepository.InsertContact(contact);
    contactRepository.InsertContact(c);
    contactRepository.InsertContact(c1);
    messageRepository.InsertChat(m);
    messageRepository.InsertChat(m1);
    messageRepository.InsertChat(m2);
}

```

```
messageRepository.InsertChat(m3);  
messageRepository.InsertChat(m4);  
messageRepository.InsertChat(m5);  
messageRepository.InsertChat(m6);  
var result = messageRepository.GetMessagesOrdered();  
Assert.AreEqual(result[0].Mensaje, "holii");  
    }  
}
```

# 11. Código e instalación

Para poder utilizar la aplicación DicereGram, no se debe tener instalada la aplicación Telegram en el dispositivo móvil. Esto es así debido a que si Telegram se encuentra instalado en el dispositivo, será dicha aplicación la que recibirá el código de verificación solicitado al iniciar la aplicación DicereGram y no será recibido a través de SMS, que es lo que se espera.

La instalación de la aplicación DicereGram se realizará a través de Google Play Store<sup>36</sup>.

Para realizar el despliegue en Google Play Store en primer lugar se ha modificado el nombre del paquete a “com.pfrias.DicereGram”.

Tras esto, compilamos el proyecto en Release.

Una vez realizado lo anteriormente descrito, pulsamos botón derecho sobre el proyecto, y, a continuación, sobre Archivar.

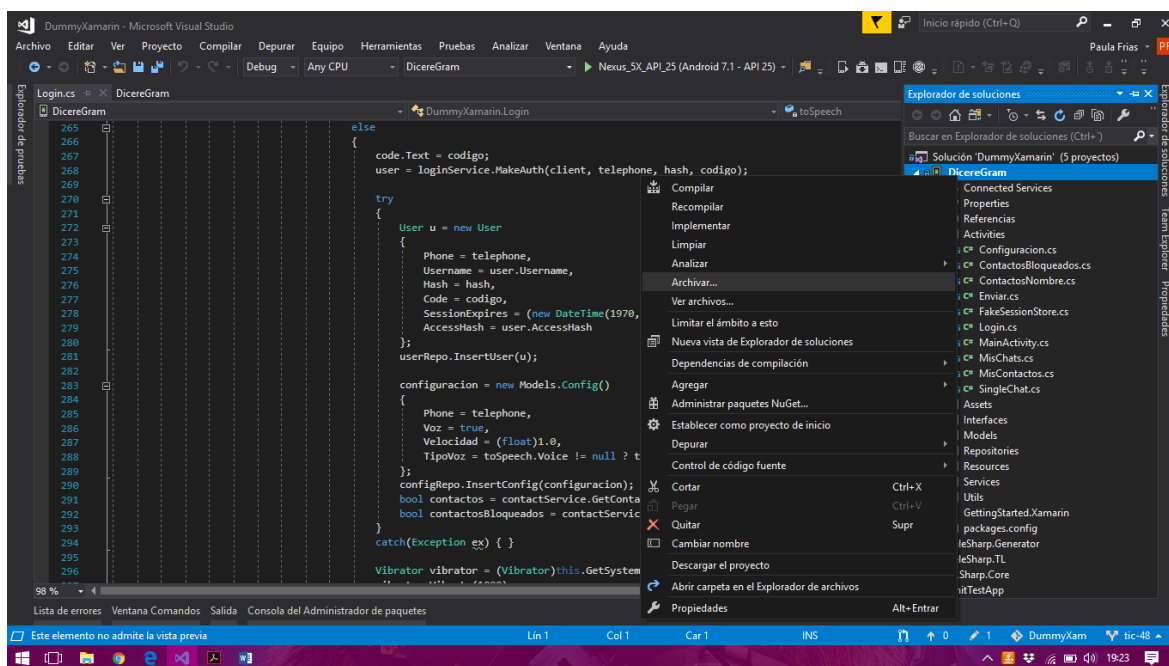


Ilustración 61 - Abrir Archive Manager Visual Studio 2017

Tras realizar el paso anterior, se nos mostrará la pantalla de “Archive Manager”. Esta herramienta de administración nos permite publicar directamente la aplicación en Google Play o realizar la publicación de manera manual.

<sup>36</sup> Google Play Store. [Sitio Web](#).

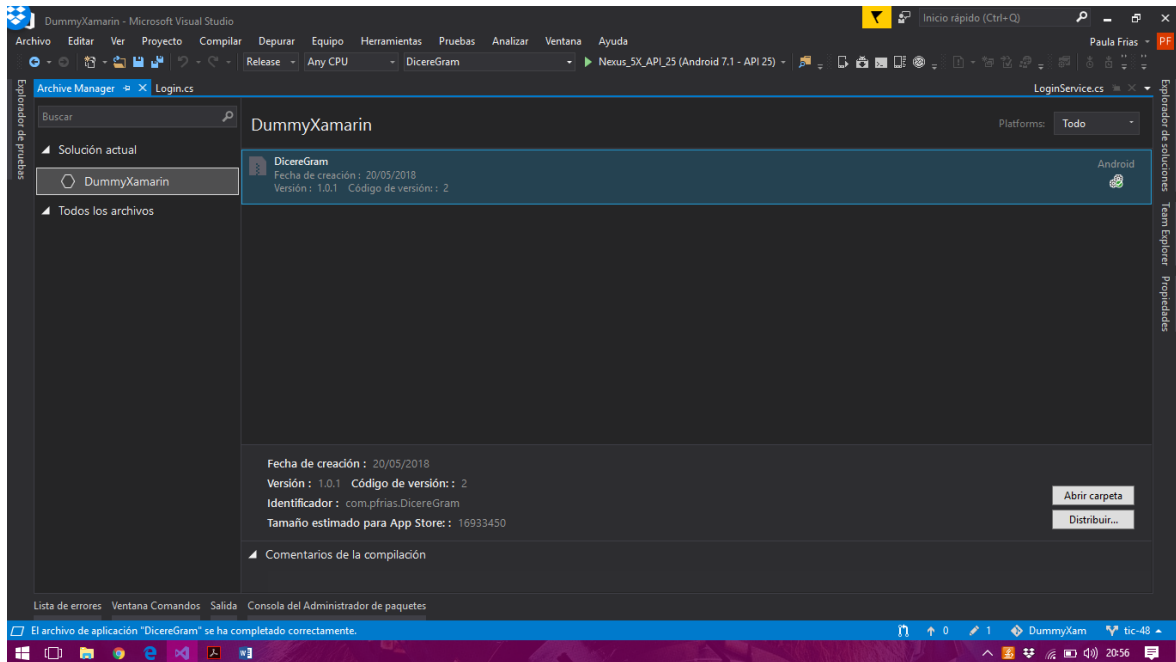


Ilustración 62 - Archive Manager Visual Studio 2017

Al ser la primera vez que vamos a distribuir la aplicación en Google Play Store, debemos hacerlo manualmente. Para ello, pulsamos sobre el botón “Distribuir”.

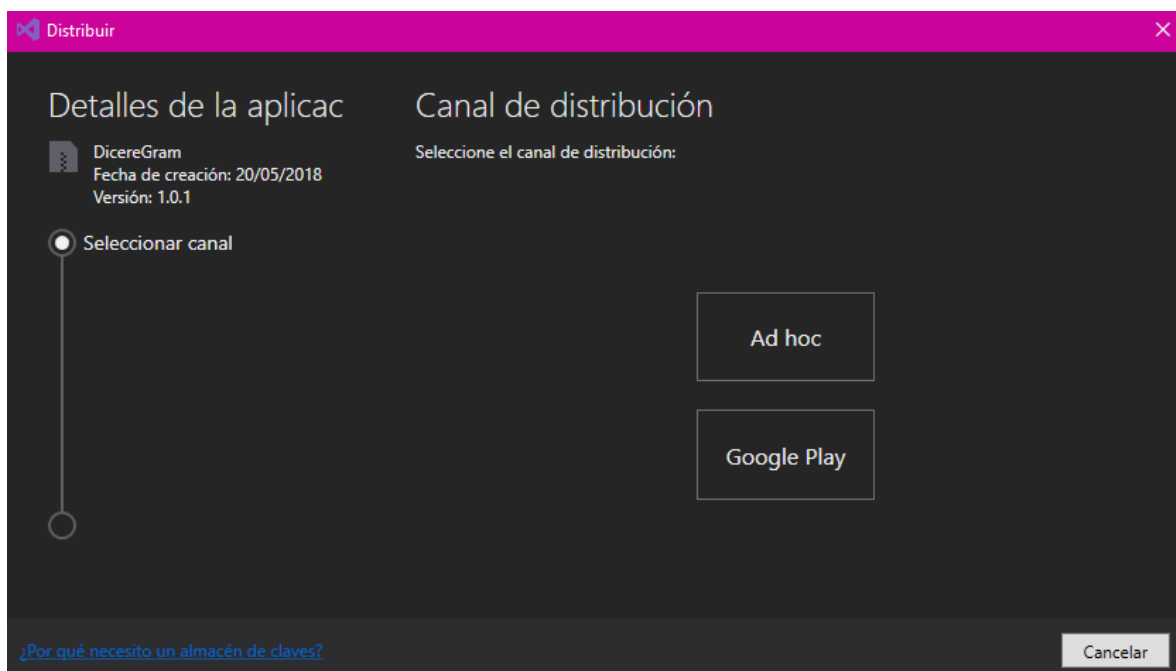


Ilustración 63 - Distribución aplicación Visual Studio 2017

Se nos mostrará el cuadro de diálogo Distribuir y pulsaremos sobre el botón “Ad-hoc”. Esta acción nos abrirá un cuadro de diálogo dónde deberemos seleccionar o crear una “identidad de firma” para poder firmar la aplicación.

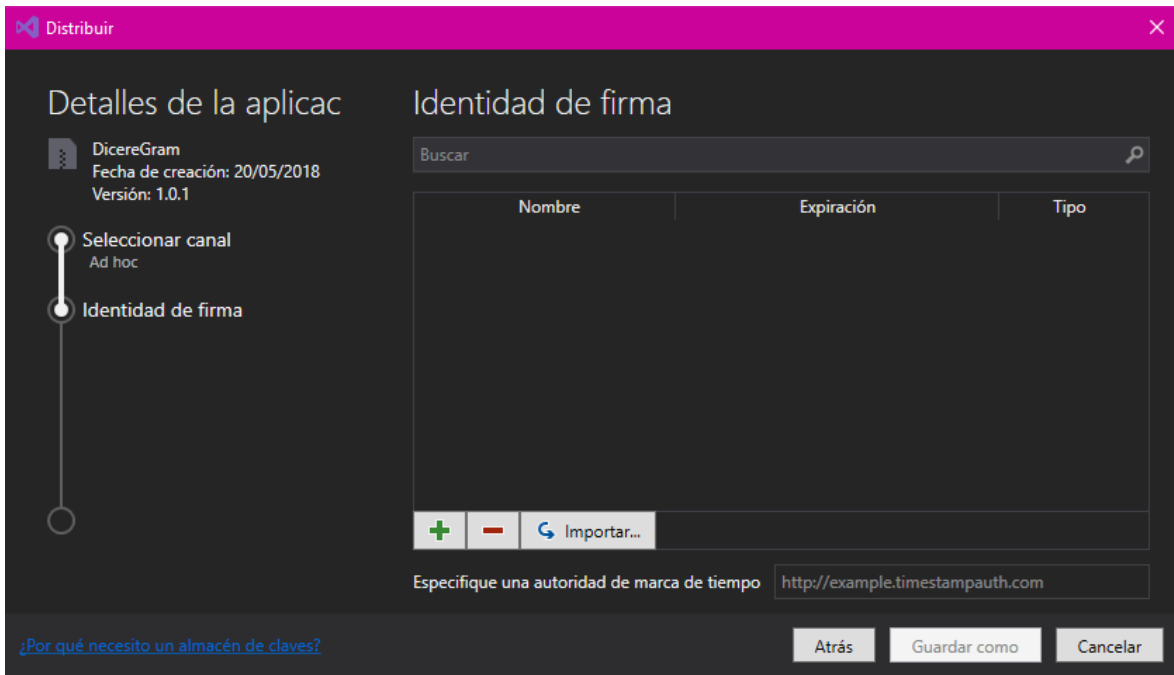


Ilustración 64 - Firma aplicación Visual Studio 2017

Añadiremos una firma, si no tenemos creada ninguna, mediante el botón “+”. Se nos mostrará el cuadro de diálogo siguiente en el que deberemos rellenar el formulario que contiene y creamos la clave de Android.

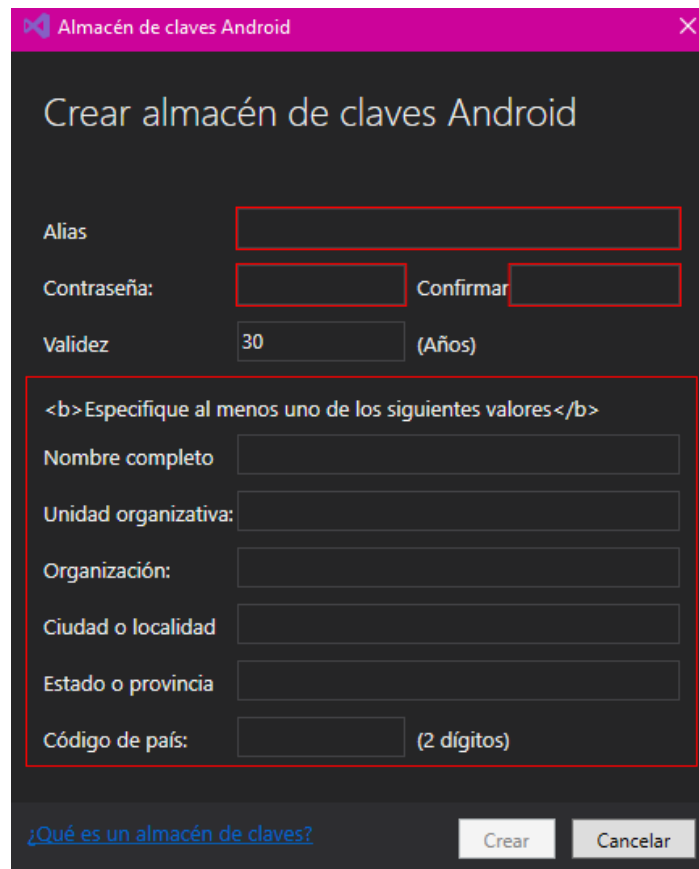


Ilustración 65 - Crear firma para aplicación Android



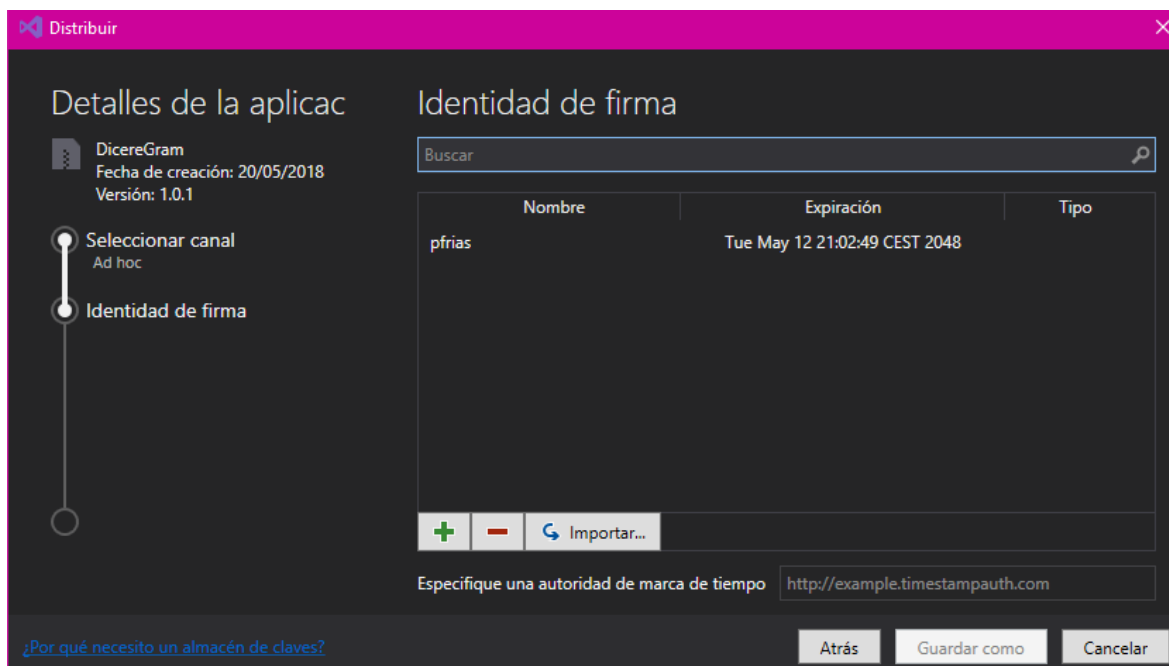


Ilustración 66 - Selección firma

A continuación seleccionamos la firma creada, y pulsamos sobre el botón “Guardar como”. Una vez realizado esto, y almacenado el fichero APK localmente, accederemos a Google Play Console<sup>37</sup>.

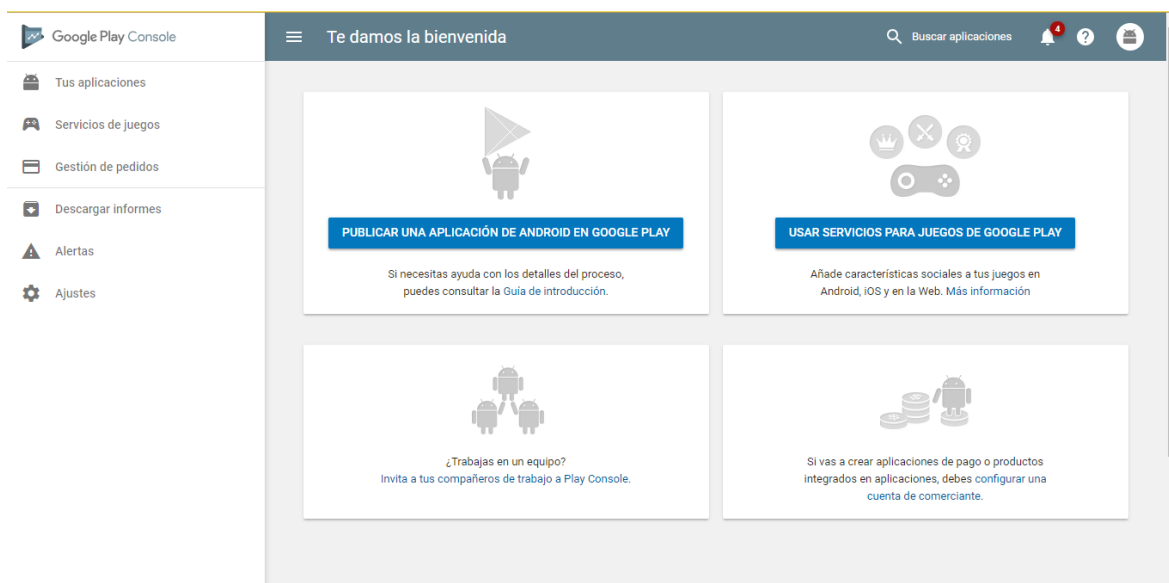


Ilustración 67 - Página principal Google Play Console

Pulsaremos sobre el menú “Tus aplicaciones” para posteriormente pulsar sobre el botón “Crear una aplicación” y rellenamos los datos que nos solicitan.

## Crear una aplicación

Idioma predeterminado \*

Español (España) – es-ES

Nombre \*

DicereGram

10/50

CANCELAR

CREAR

Ilustración 68 - Primer formulario crear aplicación

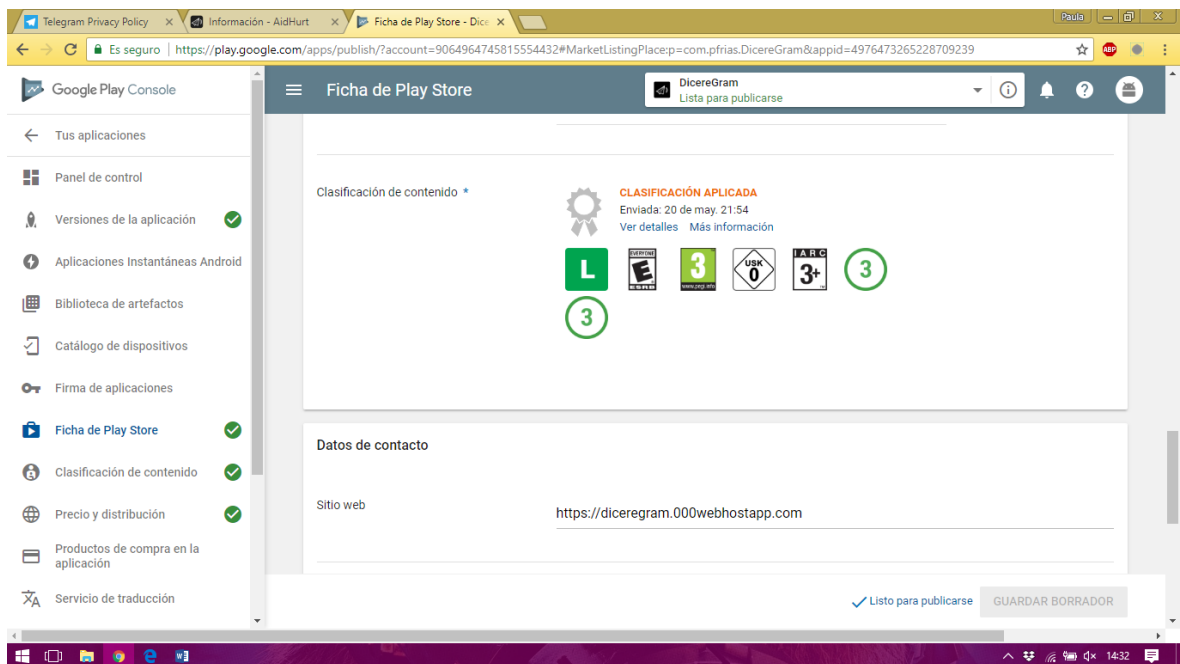
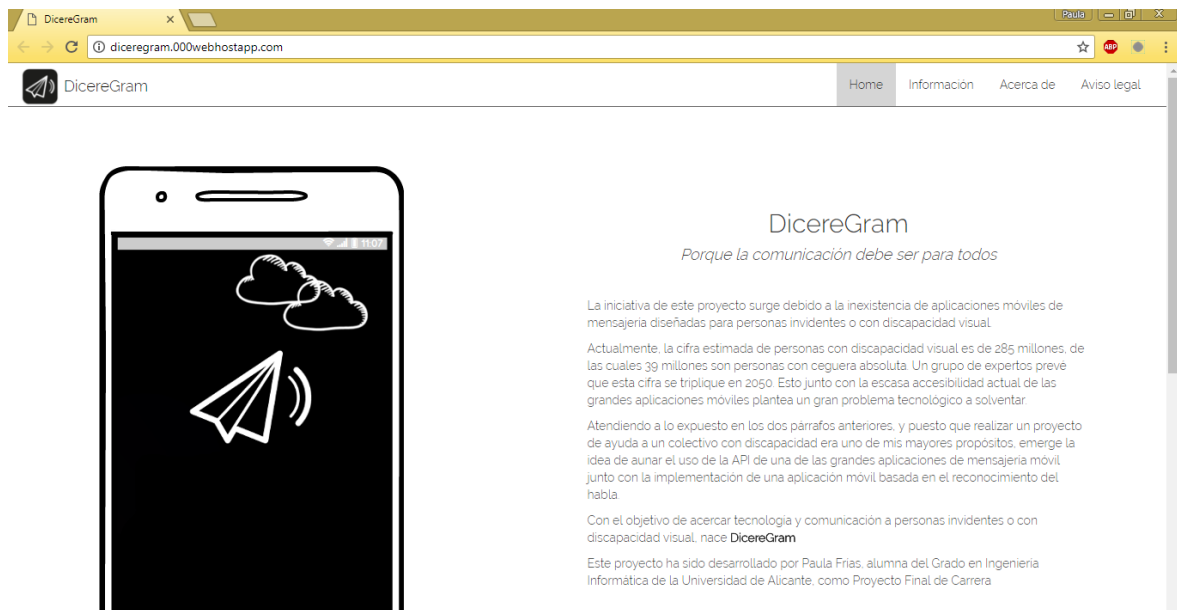


Ilustración 69 - Ficha Play Store

Una vez rellenado el formulario, pulsamos sobre “Guardar borrador” y desde el menú de la izquierda nos desplazaremos a “Versiones de la aplicación”. A continuación

pulsaremos sobre el botón “Manage” de la sección “Segmento de producción” y crearemos una nueva versión.

Para poder lanzar una aplicación que se encuentra dentro de la categoría de “comunicación”, hemos tenido que implementar una página web con una política de privacidad para informar del uso de los datos de los usuarios por parte de la aplicación DicereGram. La URL del sitio web es: <http://diceregram.000webhostapp.com>



*Ilustración 70 - Página web DicereGram*

Una vez realizados los pasos anteriores, navegaremos a la sección de “Versiones de la aplicación”. Y pulsaremos sobre el botón “Crear versión”.

En el siguiente paso, subimos nuestro fichero APK a la plataforma y almacenamos los cambios.

Para la siguiente tarea, pulsaremos sobre el ítem de menú “Clasificación de contenido” en la que rellenaremos un cuestionario.

Una vez finalizado el cuestionario y pulsado sobre los botones correspondientes para terminar la clasificación, pulsaremos sobre el ítem de menú “Precio y distribución” y rellenamos el cuestionario que se nos muestra.

Tras todo lo anterior, volvemos a la sección de “Versiones de la aplicación”. Como antes ya habíamos creado una versión, nos aparecerá un ítem aún no lanzado.

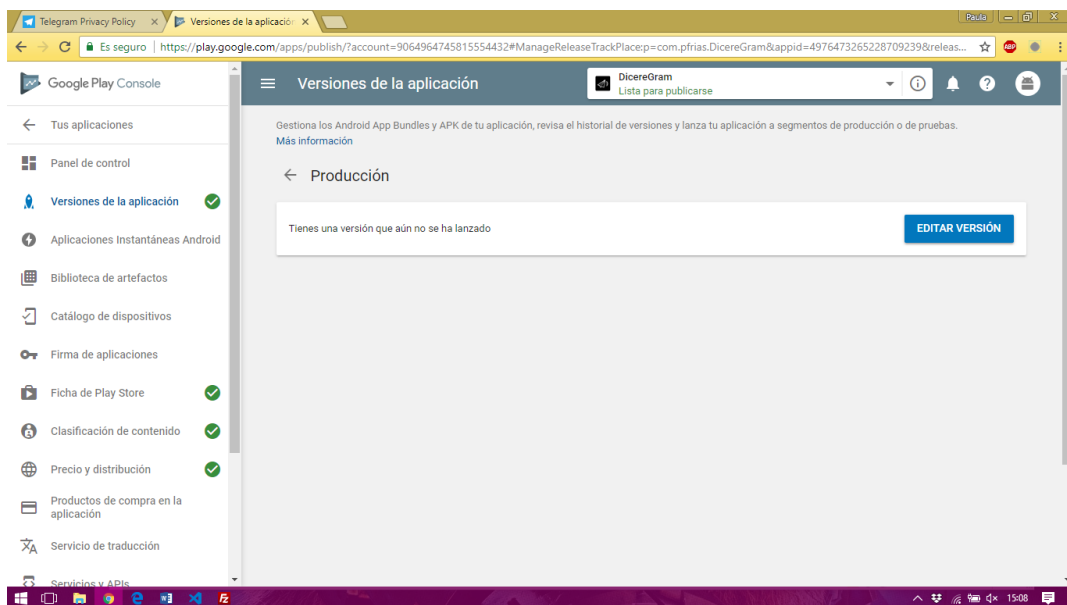


Ilustración 71 - Edición versión aplicación

Pulsamos sobre el botón “Editar versión”. Tras esto pulsaremos sobre “Revisar” (ya que el fichero apk se añadió en pasos anteriores).

Y, por último, pulsaremos sobre el botón “Iniciar lanzamiento a producción”.

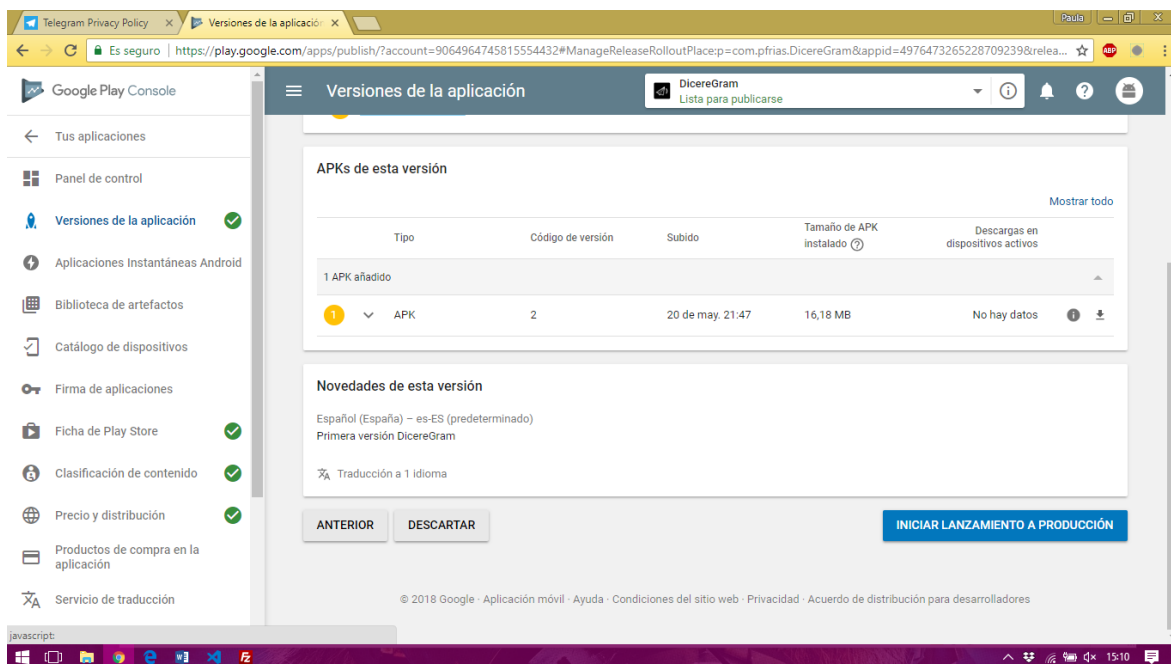


Ilustración 72 - Lanzamiento versión aplicación

Una vez realizado esto, y pasadas unas horas, la aplicación estará publicada en Google Play Store. La aplicación DicereGram está publicada y se puede acceder pulsando la siguiente imagen:



## 12. Conclusiones y mejoras

---

Una vez finalizado el proyecto, se ha realizado una retrospectiva global con el objetivo de extraer conclusiones y mejoras que podemos realizar sobre el proyecto DicereGram y que le den un mayor valor a la aplicación, cabe destacar dos importantes mejoras:

- Añadir al proyecto alguna API de reconocimiento de voz como Wit.ai
- Ampliar las funcionalidades del proyecto como por ejemplo añadir la funcionalidad completa para el uso de grupos desde la aplicación.

# 13. Bibliografía y referencias

---

49th Hawaii International Conference on System Sciences, 2016. Developing Apps for Visually Impaired People: Lessons Learned from Practice

Proyectos:

- Xamarin:
  - Repositorio: <https://github.com/pfa13/DicereGramTFG>
  - Sitio web: <https://diceregram.000webhostapp.com/>
  - Google Play Store:  
<https://play.google.com/store/apps/details?id=com.pfrias.DicereGram>
- Kotlin:
  - Repositorio: <https://github.com/pfa13/DemoKotlin>
- Api Rest C#:
  - Repositorio: <https://github.com/pfa13/DicereGramAPI>

Herramientas y tecnologías utilizadas:

- GitHub: <https://github.com/>
- Trello: <https://trello.com/>
- Telegram:
  - Sitio Web: <https://telegram.org/>
  - API: <https://core.telegram.org/api>
- TLSharp: <https://github.com/sochix/TLSharp>
- Xamarin: <https://www.xamarin.com/>
- Visual Studio 2017: <https://www.visualstudio.com/es/>
- SQLite: <https://www.sqlite.org/index.html>
- Google Play Store: <https://play.google.com/store>
- Scrum:  
<https://books.google.es/books?hl=es&lr=&id=6pZCAwAAQBAJ&oi=fnd&pg=PT9&dq=scrum&ots=kbqSXZ5qgX&sig=Ci6RE2h2uDAgGF5olejNHDEWgKQ#v=onepage&q=scrum&f=false>
- Kanban: <https://es.atlassian.com/agile/kanban>
- GitFlow: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- Android TTS:  
<https://developer.android.com/reference/android/speech/tts/TextToSpeech>

- Android Speech Recognizer:  
<https://developer.android.com/reference/android/speech/SpeechRecognizer>
- Android Gesture Detector:  
<https://developer.android.com/reference/android/view/GestureDetector>
- Balsamiq: <https://balsamiq.com/>
- NUnit: <http://nunit.org/>

Material complementario (desarrollo e implementación de código):

- BroadcastReceiver y servicios:  
[https://books.google.es/books?id=3fSoCwAAQBAJ&pg=PA232&lpg=PA232&dq=localbroadcastmanager+xamarin&source=bl&ots=pyFttDNrGA&sig=XkZ61Cx\\_SKSG7LHsSldUqOltkrQ&hl=es&sa=X&ved=0ahUKEWjKqODa9cvaAhUFD5oKHXtYBHQQ6AEIYDAG#v=onepage&q=localbroadcastmanager%20xamarin&f=false](https://books.google.es/books?id=3fSoCwAAQBAJ&pg=PA232&lpg=PA232&dq=localbroadcastmanager+xamarin&source=bl&ots=pyFttDNrGA&sig=XkZ61Cx_SKSG7LHsSldUqOltkrQ&hl=es&sa=X&ved=0ahUKEWjKqODa9cvaAhUFD5oKHXtYBHQQ6AEIYDAG#v=onepage&q=localbroadcastmanager%20xamarin&f=false)
- Android TTS: <https://docs.microsoft.com/es-es/xamarin/xamarin-forms/app-fundamentals/dependency-service/text-to-speech>
- Android Speech Recognition: <https://docs.microsoft.com/es-es/xamarin/android/platform/speech>
- Android Gesture Detector: <https://www.c-sharpcorner.com/article/creating-a-gesture-listener-in-xamarin-android-app-using-visual-studio/>
- SQLite: <https://somostechies.com/persistencia-de-datos-en-xamarin-usando-sqlite/#.WwxlKEjRDDd>
- NUnitLite:
  - <https://books.google.es/books?id=JqqVAgAAQBAJ&pg=PT101&lpg=PT101&dq=nunitlite+xamarin+android&source=bl&ots=hd1KYjxCAQ&sig=42PI5avwOHxCZkYli52xNA3CuaE&hl=es&sa=X&ved=0ahUKEwiP6qnViKnAbAhWOCuwKHZ94BhkQ6AEIbzAl#v=onepage&q=nunitlite%20xamarin%20android&f=false>
  - <https://books.google.es/books?id=3bVrCgAAQBAJ&pg=PA198&lpg=PA198&dq=nunitlite+xamarin+android&source=bl&ots=XiuoT6SHmH&sig=YgiSO2XRNNajSiA-Yv3he6gVmqg&hl=es&sa=X&ved=0ahUKEwiP6qnViKnAbAhWOCuwKHZ94BhkQ6AEIdDAJ#v=onepage&q=nunitlite%20xamarin%20android&f=false>

# 14. Anexos

---

## 14.1. Interacción conversaciones

### 14.1.1. Menú principal

#### Sexta posibilidad

<b>DicereGram</b>	¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?
<b>Usuario</b>	Cerrar sesión
<b>DicereGram</b>	¿Está seguro que quiere cerrar la sesión? Hasta dentro de 2 horas no podrá volverse a logear con el mismo número de teléfono.
<b>Usuario</b>	Sí
<b>DicereGram</b>	Ha ocurrido un problema al cerrar la sesión. Por favor, inténtelo más tarde. ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, o no hacer nada?

#### Séptima posibilidad

<b>DicereGram</b>	¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?
<b>Usuario</b>	Cerrar sesión
<b>DicereGram</b>	¿Está seguro que quiere cerrar la sesión? Hasta dentro de 2 horas no podrá volverse a logear con el mismo número de teléfono.
<b>Usuario</b>	Sí
<b>DicereGram</b>	Ha ocurrido un problema al cerrar la sesión en Telegram. Por favor, inténtelo más tarde. ¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, o no hacer nada?

#### Octava posibilidad

<b>DicereGram</b>	¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?
<b>Usuario</b>	Cerrar sesión
<b>DicereGram</b>	¿Está seguro que quiere cerrar la sesión? Hasta dentro de 2 horas no podrá volverse a logear con el mismo número de teléfono.
<b>Usuario</b>	No
<b>DicereGram</b>	¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, o no hacer nada?
<b>Usuario</b>	Nada

#### Novena posibilidad

<b>DicereGram</b>	¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, cerrar sesión, o no hacer nada?
<b>Usuario</b>	Nada
<b>Usuario</b>	El usuario pulsa sobre el botón "Logout"
<b>DicereGram</b>	¿Está seguro que quiere cerrar la sesión?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Qué quiere hacer?. ¿Enviar, leer, contactos, configuración, o no hacer nada?
<b>Usuario</b>	Nada



## 14.1.2. Enviar mensaje

### Quinta posibilidad

<b>DicereGram</b>	Diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Lo siento, no le he entendido. Por favor, diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pablo
<b>DicereGram</b>	¿Qué mensaje quiere enviar?
<b>Usuario</b>	Hola, qué tal estás
<b>DicereGram</b>	Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	Mensaje Enviado
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### Sexta posibilidad

<b>DicereGram</b>	Diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pepe
<b>DicereGram</b>	Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea enviar un mensaje, diga terminar
<b>Usuario</b>	Terminar
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### Séptima posibilidad

<b>DicereGram</b>	Diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde.
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### Octava posibilidad

<b>DicereGram</b>	Diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pablo
<b>DicereGram</b>	¿Qué mensaje quiere enviar?
<b>Usuario</b>	Hola, qué tal estás
<b>DicereGram</b>	Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	Ha ocurrido un error al enviar el mensaje. Por favor, inténtelo más tarde.
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### Novena posibilidad

<b>DicereGram</b>	Diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pablo
<b>DicereGram</b>	¿Qué mensaje quiere enviar?
<b>Usuario</b>	Hola, qué tal estás
<b>DicereGram</b>	Su mensaje es: Hola, qué tal estás. ¿Quiere enviarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde.
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### 14.1.3. Menú contactos

#### Quinta posibilidad

<b>DicereGram</b>	¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?
<b>Usuario</b>	Eliminar
<b>DicereGram</b>	Diga el nombre del contacto que quiere eliminar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto a quien quiere enviar un mensaje
<b>Usuario</b>	Pablo
<b>DicereGram</b>	¿Está seguro que quiere eliminar a Pablo?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (no hay voz)

#### Sexta posibilidad

<b>DicereGram</b>	¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?
<b>Usuario</b>	Eliminar
<b>DicereGram</b>	Diga el nombre del contacto que quiere eliminar
<b>Usuario</b>	Pepe
<b>DicereGram</b>	Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea eliminar, diga terminar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto que desea eliminar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	¿Está seguro que quiere eliminar a Pablo?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (no hay voz)

#### Séptima posibilidad

<b>DicereGram</b>	¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?
<b>Usuario</b>	Eliminar
<b>DicereGram</b>	Diga el nombre del contacto que quiere eliminar
<b>Usuario</b>	Pepe
<b>DicereGram</b>	Lo siento, no existe ese contacto. Por favor, diga el nombre de un contacto válido o si no desea eliminar, diga terminar
<b>Usuario</b>	Terminar
<b>DicereGram</b>	¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (no hay voz)

#### Octava posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Bloqueados

**DicereGram** Redirección a ContactosBloqueados (no hay voz)

#### Novena posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Nada

#### Décima posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Volver atrás (o atrás o volver)

**DicereGram** Redirección a MainActivity (no hay voz)

#### Undécima posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** x

**DicereGram** Lo siento, no ha dicho una opción válida. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Volver atrás (o atrás o volver)

**DicereGram** Redirección a MainActivity (no hay voz)

#### Duodécima posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Actualizar

**DicereGram** Ha ocurrido un error al actualizar los contactos. Por favor, inténtelo más tarde. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Volver atrás (o atrás o volver)

**DicereGram** Redirección a MainActivity (no hay voz)

#### Decimotercera posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Eliminar

**DicereGram** Diga el nombre del contacto que quiere eliminar

**Usuario** Pablo

**DicereGram** Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Volver atrás (o atrás o volver)

**DicereGram** Redirección a MainActivity (no hay voz)

#### Decimocuarta posibilidad

**DicereGram** ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Eliminar

**DicereGram** Diga el nombre del contacto que quiere eliminar

**Usuario** Pablo

**DicereGram** ¿Está seguro que quiere eliminar a Pablo?

**Usuario** Sí

**DicereGram** Ha ocurrido un error al eliminar el contacto. Por favor, inténtelo más tarde. ¿Quiere buscar un contacto, actualizar, eliminar, bloqueados, no hacer nada o volver atrás?

**Usuario** Volver atrás (o atrás o volver)

**DicereGram** Redirección a MainActivity (no hay voz)

## 14.1.4. Buscar contacto

### Sexta posibilidad

<b>DicereGram</b>	Diga el nombre del contacto que quiere buscar
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento, no le he entendido. ¿Quiere buscar un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás (o volver o volver atrás)
<b>DicereGram</b>	Redirección a MisContactos (no hay voz)

### Séptima posibilidad

<b>DicereGram</b>	Diga el nombre del contacto que quiere buscar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	El contacto buscado es: Nombre: Pablo Corral. Teléfono: x. Estado: x. Bloqueado: No está bloqueado.
<b>DicereGram</b>	¿Quiere buscar otro contacto, no hacer nada o volver atrás?
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento, no ha dicho una opción válida. ¿Quiere buscar un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás (o volver o volver atrás)
<b>DicereGram</b>	Redirección a MisContactos (no hay voz)

### Octava posibilidad

<b>DicereGram</b>	Diga el nombre del contacto que quiere buscar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere buscar un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás (o volver o volver atrás)
<b>DicereGram</b>	Redirección a MisContactos (no hay voz)

### Novena posibilidad

<b>DicereGram</b>	Diga el nombre del contacto que quiere buscar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto que desea buscar
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere buscar un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás (o volver o volver atrás)
<b>DicereGram</b>	Redirección a MisContactos (no hay voz)

## 14.1.5. Bloquear/Desbloquear un contacto

### Quinta posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento, no le he entendido. ¿Quiere buscar un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Sexta posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás (o volver o atrás)
<b>DicereGram</b>	Redirección a MisContactos (no hay voz)

### Séptima posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento, no ha dicho una opción válida. ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?";
<b>Usuario</b>	Nada

### Octava posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Bloquear
<b>DicereGram</b>	Diga el nombre del contacto que quiere bloquear
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un problema al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Novena posibilidad

<b>DicereGram</b>	Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Bloquear
<b>DicereGram</b>	Diga el nombre del contacto que quiere bloquear
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto que desea bloquear/desbloquear
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un problema al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Décima posibilidad

**DicereGram** Sus contactos bloqueados son: .... ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?

**Usuario** Bloquear

**DicereGram** Diga el nombre del contacto que quiere bloquear

**Usuario** Pablo

**DicereGram** Existen 2 contactos que contienen ese nombre: Dani Pablo y Pablo. Diga el nombre del contacto que desea bloquear/desbloquear

**Usuario** Pablo

**DicereGram** Ha ocurrido un problema bloquear o desbloquear el contacto. Por favor, inténtelo más tarde. ¿Quiere bloquear un contacto, desbloquear un contacto, no hacer nada o volver atrás?

**Usuario** Nada

## 14.1.6. Mensajes general

### Quinta posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Borrar
<b>DicereGram</b>	Por favor, diga el nombre del contacto
<b>Usuario</b>	Pepe
<b>DicereGram</b>	Lo siento, no existe conversación con Pablo. Diga el nombre de un contacto válido.
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Conversación con Pablo borrada. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?

### Sexta posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Séptima posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### Octava posibilidad

<b>DicereGram</b>	No tiene mensajes nuevos. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a MainActivity (no hay voz)

### Novena posibilidad

<b>DicereGram</b>	Ha ocurrido un error al obtener sus mensajes nuevos. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Décima posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento, no ha dicho una opción válida. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás
<b>DicereGram</b>	Redirección a Menú principal (no hay voz)



#### Undécima posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Entrar/borrar
<b>DicereGram</b>	Por favor, diga el nombre del contacto para entrar a su conversación
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un error al entrar/borrar. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás
<b>DicereGram</b>	Redirección a Menú principal (no hay voz)

#### Duodécima posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Entrar/borrar
<b>DicereGram</b>	Por favor, diga el nombre del contacto para entrar a su conversación
<b>Usuario</b>	Pablo
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Atrás
<b>DicereGram</b>	Redirección a Menú principal (no hay voz)

#### Decimotercera posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Leerlos
<b>DicereGram</b>	Los mensajes de Pablo son... Los mensajes de Cristina son... Los mensajes de Zoe son... ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

#### Decimocuarta posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Leerlos
<b>DicereGram</b>	Hay demasiados mensajes sin leer. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

#### Decimoquinta posibilidad

<b>DicereGram</b>	Tienes 9 mensajes nuevos de Pablo, Cristina y Zoe. ¿Quiere leerlos, entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Leerlos
<b>DicereGram</b>	Ha ocurrido un error al leer los mensajes. ¿Quiere entrar a una conversación, borrarla, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

## 14.1.7. Mensajes chat individual

### Sexta posibilidad

<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Buscar por mensaje
<b>DicereGram</b>	¿Qué mensaje quiere buscar?
<b>Usuario</b>	Xamarin
<b>DicereGram</b>	No existen mensajes que concuerden con su mensaje. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Séptima posibilidad

<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento no le he entendido. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a MisChats (no hay voz)

### Octava posibilidad

<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Novena posibilidad

<b>DicereGram</b>	Los mensajes de Pablo son... ¿Quiere continuar leyendo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	... ¿Quiere continuar leyendo?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Décima posibilidad

<b>DicereGram</b>	¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	x
<b>DicereGram</b>	Lo siento, no ha dicho una opción válida. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a MisChats (no hay voz)

### Undécima posibilidad

<b>DicereGram</b>	Los mensajes nuevos de Pablo son: ... ¿Quiere responder?
<b>Usuario</b>	Sí
<b>DicereGram</b>	¿Qué mensaje quiere enviar?
<b>Usuario</b>	Holi
<b>DicereGram</b>	Su mensaje es Holi. ¿Quiere enviarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	No se ha podido enviar el mensaje. Inténtelo más tarde. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Duodécima posibilidad

**DicereGram** ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?  
**Usuario** Mensaje  
**DicereGram** Diga el mensaje desde la que quiere leer  
**Usuario** Xamarin  
**DicereGram** Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde  
¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?  
**Usuario** Volver atrás  
**DicereGram** Redirección a MisChats (no hay voz)

### Decimotercera posibilidad

**DicereGram** Los mensajes nuevos de Pablo son: ... ¿Quiere responder?  
**Usuario** Sí  
**DicereGram** ¿Qué mensaje quiere enviar?  
**Usuario** Holi  
**DicereGram** Su mensaje es Holi. ¿Quiere enviarlo?  
**Usuario** Sí  
**DicereGram** Ha ocurrido un error al enviar el mensaje. Inténtelo más tarde. ¿Quiere leer desde una fecha, buscar por un mensaje, no hacer nada o volver atrás?  
**Usuario** Nada

## 14.1.8. Menú configuración

### Cuarta posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Modificar nombre de usuario
<b>DicereGram</b>	Su nombre de usuario es X. ¿Quiere modificarlo?
<b>Usuario</b>	Sí
<b>DicereGram</b>	¿Qué nombre de usuario quiere?
<b>Usuario</b>	paufri
<b>DicereGram</b>	Ha ocurrido un error al modificar su nombre de usuario. Por favor, inténtelo más tarde. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura o la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Quinta posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Tipo de voz
<b>DicereGram</b>	La voz actual es X. Las voces actualmente instaladas son Z y T. Diga el nombre de la voz a cambiar o diga cancelar
<b>Usuario</b>	Z
<b>DicereGram</b>	Voz modificada. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura o la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Sexta posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Tipo de voz
<b>DicereGram</b>	La voz actual es X. Las voces actualmente instaladas son Z y T. Diga el nombre de la voz a cambiar o diga cancelar
<b>Usuario</b>	Cancelar
<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura o la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

### Séptima posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Tipo de voz
<b>DicereGram</b>	La voz actual es X. Las voces actualmente instaladas son Z y T. Diga el nombre de la voz a cambiar o diga cancelar
<b>Usuario</b>	T
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura o la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

#### Octava posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Tipo de voz
<b>DicereGram</b>	La voz actual es X. No tiene más voces instaladas. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura o la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

#### Novena posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Velocidad
<b>DicereGram</b>	La velocidad de lectura es normal. ¿Quiere aumentar la velocidad de lectura, reducirla o cancelar?
<b>Usuario</b>	Aumentar/Reducir
<b>DicereGram</b>	Velocidad de lectura aumentada/reducida. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

#### Décima posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Velocidad
<b>DicereGram</b>	La velocidad de lectura es normal. ¿Quiere aumentar la velocidad de lectura, reducirla o cancelar?
<b>Usuario</b>	Aumentar/Reducir
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Nada

#### Undécima posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Velocidad
<b>DicereGram</b>	La velocidad de lectura es super rápida. ¿Quiere aumentar la velocidad de lectura, reducirla o cancelar?
<b>Usuario</b>	Aumentar
<b>DicereGram</b>	No se puede aumentar más la velocidad de lectura. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (sin voz)

#### Duodécima posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Activación
<b>DicereGram</b>	La voz está activada/desactivada. ¿Quiere desactivarla/activarla?
<b>Usuario</b>	Sí
<b>DicereGram</b>	Voz activada/desactivada. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (sin voz)

#### Decimotercera posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Activación
<b>DicereGram</b>	La voz está activada/desactivada. ¿Quiere desactivarla/activarla?
<b>Usuario</b>	No
<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (sin voz)

#### Decimocuarta posibilidad

<b>DicereGram</b>	¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Activación
<b>DicereGram</b>	La voz está activada/desactivada. ¿Quiere desactivarla/activarla?
<b>Usuario</b>	Sí
<b>DicereGram</b>	Ha ocurrido un error al acceder a la base de datos. Por favor, inténtelo más tarde. ¿Quiere modificar su nombre de usuario, el tipo de voz, la velocidad de lectura, la activación de la voz, no hacer nada o volver atrás?
<b>Usuario</b>	Volver atrás
<b>DicereGram</b>	Redirección a Menú principal (sin voz)